国際セミナーTEI Day in Kyoto 2006報告書

京都大学21世紀COEプログラム

東アジア世界の人文情報学研究教育拠点

漢字文化の全き継承と発展のために

国際セミナー　TEI Day in Kyoto 2006
報告書

2006年5月17日(水)

京都大学 百周年時計台記念館

拠点リーダー　高田時雄

(京都大学人文科学研究所)

2006年12月

前書き

この小冊子は，2006 年 5 月 17 日に京都大学百周年時計台記念館で開催された，TEI Day in Kyoto 2006 の報告書です．この会議は，京都大学の 21 世紀 COE プログラム「東アジア世界の人文情報学研究教育拠点」の活動の一環として，TEI(Text Encoding Initiative)コンソーシアムとの協賛で開催されました．TEI は，人文情報学に関心を寄せる研究機関や人々から構成される組織で，ISO や W3C といった国際規格を策定する組織と密に連携を取りながら，人文科学で使われる様々なテキストの電子化に対応する符号・交換規格を開発しています．また，本 COE により，Unicode のような規格から外れている，滅多に使われることなく一般的ではない文字を，交換可能なものとして符号化することが可能になりました．

この会議は，TEI ガイドラインの作成に中心的な役割を果たしているメンバーと，TEI を日本やアジアで使用している人々が，共に集まることを意図して開催されました．また，この会議は，日本のみならずアジアにおいて，TEI コンソーシアムが主催・協賛する初めての会議となりました．その意味で，この会議で繰り広げられた論議は，大変に重要なものです．この小冊子で，その模様を皆さんにお伝えできればと存じます．また，当日の話では少ししか触れられていなかったことや，話には出てこなかった内容も，この小冊子には収録されています．

当日の会議は，まず主催者側から本会議の趣旨や内容について，簡単な説明がおこなわれました．続いて，TEI やマークアップ活動に携わる日本の研究者から，それぞれのテーマで発表がありました．この小冊子には，その中から，松村一登先生と大矢一志先生の論文が収録されています．残念ながら，当日の発表にあった土屋俊先生の論文は収録されておりません．

午後は，ポスターセッションから始まりました．このポスターセッションは，その日一番の盛り上がりをみせ，参加者が一番頭を使った時間となりました．当日のポスター発表の様子を知って頂くために，この小冊子には，ポスター内容のまとめ(ポスターそのものではない)と，その要旨の英語版と日本語版を収録してあります．ポスターセッションに続いて行われた，午後のセッションでは，TEI 技術委員会の 5 人のメンバー，シド・バウマン(Syd Bauman)，セバスチャン・ラッツ (Sebastian Rahtz)，マシュー・ドリスコル(Matthew Driscoll)，コナル・トーイ(Conal Tuohy)，ジェームズ・カミングス(James Cummings)の各氏から発表がありました．全ての発表が，最新の論議を扱い，刺激的なものになっていました．この小冊子には，これら 5 つの発表についての全ての論文が，発表順に収録されています．

最後に，このような機会を設けることが出来ましたことを，参加して頂きました全ての方々に感謝致します．またとりわけ，本会議の開催にご尽力を頂きました方々には，心から感謝申し上げます．

２００６年９月　　ウィッテルン　クリスティアン

Preface

This litte booklet presents a record in written form of the proceedings from the "TEI Day in Kyoto 2006", held at Kyoto University's Clock Tower Centennial Hall on May 17th, 2006, organized as part of the activities of the Kyoto University 21st century COE program *Toward an Overall Inheritance and Development of Kanji Culture*, co-sponsored by the TEI Consortium. The TEI is an international membership consortium of stakeholders in the Digital Humanities, developing, in close cooperation with international standard bodies like the ISO and the World Wide Web-Consortium, a standard for encoding and interchange of electronic text in all fields of the Humanities. Thanks also to this COE program, it has recently become possible to encode rare and unusual characters in a standardized, exchangeable way, even if they are not part of standardized character sets such as Unicode.

This event was designed to bring together the leading developers of the *TEI Guidelines for Electronic Text Encoding and Interchange* as represented by the current members of the TEI Technical Council and practitioners of TEI in Japan and its Asian neighbours. It was also the first event (co-)sponsored by the TEI Consortium, not only in Japan, but also in all of Asia. Insofar, it was a memorable event with a significance far beyond the deliberations of the day, which this booklet hopes to preserve. Alas, only a fraction of what was presented on that day, and nothing of the lively discussions, has entered into the printed pages here.

The day started with a short introduction by the organizers, explaining why this event was organized and what activities led to it. This was followed by some presentations of Japanese scholars, concerned with adopting the TEI and markup activities in general to their respective research agendas. Two papers from this session, by Matsumura Kazuto and Ohya Kazushi are represented in this booklet, the presentations by Tutiya Syun and his team unfortunately could not be included here.

The afternoon began with what turned out to be the most buzzling and lively part of the day, the Poster Session. To record the breadth and width of the posters presented here, the abstracts from that session (not the posters themselves) are included here, together with the abstracts of all paper presentations, in both Japanese and English versions. This session was followed by the afternoon session, with papers by five current members of the TEI Technical Council: Syd Bauman, Sebastian Rahtz, Matthew Driscoll, Conal Tuohy and James Cummings; each presentation followed by a lively and thought-provoking discussion. All these papers are included here, the order is the order they were delivered in.

In conclusion, I would like to take this opportunity to thank all participants, the audience and all those who worked hard to make this event possible for their nevertiring efforts.

Christian Wittern, September 2006

Contents

文字化された言語資源の少ない言語とテクストのマークアップ

松村一登
東京大学大学院・人文社会系研究科・言語動態学専門分野

1. 言語資料

話されたことばは，一過性のものであり，録音したり，文字で書き留めたりしておかない限り，たちまち消えてしまう。書かれたことばでも，メモのように，用事がすむとすぐに破棄されるものが結構あり，出版物は別として，残るものはほんの一部であろう。私たちが一生の間に生み出すことばは膨大なものだが，そのほとんどは，このように文字通り消えてしまい，記録に残らない。

話されたものであるか，書かれたものであるかを問わず，録音，文字表記など，何らかの形で残されたことばの記録のひとつひとつを「言語資料」と呼び，ある言語の言語資料を総体として話題にするときには (X語の)「言語資源」という言い方をすることにする。日本語や英語は，言語資源の豊かな言語であるが，アイヌ語やサーミ語は言語資源が少ない言語である。

定義上，言語資料は文字化されているとはかぎらないが，本稿では，とくに区別の必要がある場合を除いて，通常「言語資料」を「文字化された言語資料」と同義で用いる。文字化された言語資料のうち，機械可読な形態のものを「電子化された言語資料」と呼ぶことにする。〔注１〕

2. 世界の言語的多様性

Ethnologue の名前で知られる Web サイト([14]) のデータによれば，現在，地球上で実際に使われている言語の総数は 6912 であるという (図１)。6900 の言語の母語話者の数はまちまちであり，母語話者人口が１千万人以上の 83 言語 (世界の言語数の 1.2%) の母語話者の合計だけで世界の総人口の８割 (79.46%) を占める一方で，

言語数の上で全体の８割強 (82.1%) を占めている母語話者人口が 10 万人に満たない言語の母語話者数の合計は，世界の総人口のわずか 1.16%にしか達しない。

世界の言語数は，1950 年代には 2000〜3000 ([5])と考えられていたらしいから，単純に数字だけを見て，半世紀の間に言語の数が２倍以上に増えたと受けとめる人がいるかもしれない。しかし，これは，交通機関の発達などにより，世界のすみずみまで調査が進んで，50 年前には知られていなかった言語が多数「発見」されたことによる見かけの増加として理解するのが適当であって，地球上の言語の数は，長期的に見ると，むしろ急激に減少し続けているという見方が支配的である。悲観的な予測によれば，世界の言語は今後，２週間に１言語程度の割合で消滅しつづけ，100 年後には，その数が半分くらいになってしまうとさえ言われている ([7], [8])。

動植物に関して「絶滅寸前種」(endangered species) が語られるように，人間の言語についても，「消滅(の危機)に瀕した言語」(languages in danger of disappearing)，あるいは「危機言語」(endangered languages) という言い方がされる ([16])。〔注２〕「Ｘ語は危機言語か」という質問をしばしば受けるが，理論上はともかく，ある言語が危機言語かどうかの客観的な基準 (例えば，話者人口)があるというよりは，「何らかの理由で１〜２世代のうちに話し手がいなくなってしまうことが危惧される言語」はすべて危機言語だと見なすのが，現場では，いちばん無難な危機言語の定義であると筆者は考えている。

話し手がいなくなって使われなくなる言語がある一方で，新しい言語が誕生しているのもまた事実である。その背景には，「言語」の定義が少しばかり変わってきたことがある。新しく生まれた言語の多くは，これまでどこかの言語の「方言」として扱われてきたものである。言語学者たちは従来，ある言語コミュニティの話す言語形態が，独立した言語であるか，あるいは，隣接する言語形態と「方言」の関係にあるかは，語形の類似の度合いや相互理解の可能性の度合いなど，外から観察できる客観的な要因によって決めることができると考えてきた。しかし，最近は「言語」であるか「方言」であるかを決める際には，言語コミュニティの話者の意

識という主観的な要因が大きな役割を果たすという考え方が主流になりつつある。それゆえ，伝統的には何らかの言語の「方言」とされている言語形態でも，危機言語と呼ぶことがしばしば行われている。

地球上の言語の世界で一種の新陳代謝が行われているだけで，少数言語の盛衰をいちいち気に掛ける必要はないと考える人がいるかもしれない。だからといって，話し手がいなくなってこの地上から消えてゆく言語がまったく記録されないまま消えていってもかまわないということにはならない。危機言語は，使われなくなってしまわないうちに，しかるべき方法で記録にとどめておくべきである。危機言語の言語資料を記録にとどめることを目的とする言語学者たちの研究活動は「言語の記録」(language documentation)，あるいは「記録言語学」(documentary linguistics) などと呼ばれる ([11], [15])。このほか，危機言語を再活性化 (revitalization) させようとする活動も各地で行われている。

3. 文字化された言語資源の少ない言語

数の上で世界の言語の８割をしめる母語話者人口が 10 万人に満たない少数言語のほとんどは，上に述べた危機言語のカテゴリに属すると考えてよいと思われる。これらの言語は，ほとんどの場合，日常的な場面で話す言語として用いられるのみで，文字言語として使われる伝統をもたない，いわゆる「文字のない言語」であると考えられるから，文字化された言語資料がないのは，当たり前と言えば当たり前である。

このような言語すべてが，文字化された言語資料をまったく残さずに消滅してしまう運命にあると言い切るのは言い過ぎである。たとえば，言語学者などが，音声表記 (phonetic transcription)を用いて文字転写し，注釈を加えたり，翻訳を添えて，学術出版物として「X語テクスト集」のようなタイトルをつけて，少部数ながら印刷出版した言語資料が存在する言語も少なからず見られるからである。

「文字をもたない言語」の姿を，音声記号 (phonetic alphabet; cf. [6]) などによっ

て文字転写した原語テクストに，言語学的情報 (形態，品詞，意味など)を付加した言語資料の形で記録し，保存していくことを主要な目的とする研究活動が，言語の記録(language documentation) あるいは記録言語学 (documentary linguistics)という名前で，言語学の新しい分野として認知されるようになったのはここ 10 年くらいのできごとだが，言語学者たちによるこういった研究活動そのものには，音声記号の考え方と同じくらいの歴史的伝統がある。〔注３〕

この種の言語資料は，研究者ごとにさまざまなフォーマットで編集されている。例として４種類のサンプル(図２～５)を掲げる。

図２― ウデヘ語 (ツングース系)，音素表記，日本語訳 ([9] p.104)
図３― アイヌ語，音素表記，日本語訳 ([10] p.364)
図４― ベプス語 (ウラル系，ロシア)，音声表記，ロシア語訳 ([12] p.36-37)
図５― カレリア語 (ウラル系，ロシア)，音声表記，エストニア語訳 ([13] p.175)

ここで，音素表記 (phonemic transcription)と音声表記 (phonetic transcription)の区別は厳格なものではない。アイヌ語の言語資料 (図３)のように，カタカナ表記とローマ字表記が一種の正書法として定着しているケースがある一方で，それとは対照的に，音声記号と複雑な補助記号を多用して，できるかぎり精密な音声転写を行おうとする伝統の行われているウラル系諸語 (図４・図５) のようなケースもある。音素表記(＝表音文字による比較的簡略な表記)と音声表記(＝音声記号を用いたより精密な表記)の違いは，音声表記の精密さの程度の問題といっていい。

言語資料には，原文テクストに，１つ１つの単語ごとに，品詞分類や形態分析などの言語学的な情報を付加し，さらに，１つ１つの文ごとに，英語や日本語のような一般になじみのある言語への翻訳を付加するのが，言語学における標準的な慣行である。筆者が研究対象としている言語の１つ，ウラル系のマリ語の文を使って例示するならば，つぎのようになる。

tošto　jüla-m　kudalt-m-em　ok　šu　[原語]

古い　習慣-対格 廃棄する-分詞-１単　否定-３単 届く　[注解]
「私は古い慣習を捨てたくない」　[翻訳]

原語部分は，この例では，形態素レベルまで形態分析が行われているが，図３・図４・図５のように，単語レベルまでの分析 (分かち書き) しか行っていないこともある。また，注解の形式に一定の決まった方式があるわけではなく，言語ごとに，研究者ごとに，まちまちであるのがふつうである。さらには，図２～図５のように，注解が施されていない言語資料も多い。注解がない場合は，翻訳だけが付加情報として添えられているわけだが，その翻訳も，図２・図３のように，単語ごとの逐語訳，ないしは，行ごとの逐語訳に近い形で与えられることもあれば，図４・図５のように，ただ併記されていたり，対訳になっていたりするだけであることも少なくない。

注解の主体が文法的な情報であることからも明らかなように，この種のテクスト集は，もともとは，言語学者による文法研究のための一次資料として用いられることを目的として編まれたものである。筆者が専門とするウラル系バルト・フィン諸語の少数言語の場合，このようなテクスト集が，20 世紀の中頃から後半にかけて，いくつも出版されているが，これらの言語の話者が激減した現在，この種のテクスト集は，研究者たちだけでなく，当該言語のコミュニティからも，非常に貴重な文化遺産として再評価されるべきものになっている。

4. 少数言語の言語資料の電子化 ― どのような困難があるか

言語資料の電子化の前提は，それが文字転写されていることである。当然のことだが，文字言語の伝統がない言語の場合，文字転写ができる前提は，話されたことばを録音したものがあることである。文字言語の確立している日本語の場合でさえ，談話資料の文字化にはたいへんな労力 (大学院生，若手研究者など，読み書きのできる母語話者を多数動員！) と資金が必要とされることは周知の事実である。文字言語の確立していない言語の録音資料を文字転写（音声表記）する作業のためには，

音声学などの専門の訓練を受けた研究者を養成する必要がある。

本稿は，文字化されている言語資料のマークアップについて論じるのが目的なので，以下では，音声資料の収録とその文字化の作業の段階で遭遇するさまざまな問題についての議論はさけて，すでに文字転写が行われている言語資料があるものとして話をすすめる。音声資料の収録と文字化の作業の中で現れる様々な問題については，別の機会に論じることができれば幸いである。

上で実例をみたような言語資料の提示の方式は，現在，大部分の言語学者にとって，自分自身の研究のために資料を整理し保存するときのフォーマットであると同時に，印刷物として刊行される研究成果の中で言語資料を公開するときの標準的なフォーマットにもなっている。このため，言語資料を，このようなフォーマットでプリンタ出力できるように，ワープロソフトで清書したり，表計算ソフトで整理している言語学者が多いのは当然の成り行きだが，ワープロや表計算ソフトによる文書作成作業，すなわち，印刷用の完全原稿，いわゆる camera-ready の版下を容易に準備できるような書式で言語資料を整理する作業を，言語資料の電子化と同一視している (と思われる) 言語学者が圧倒的に多いように見受けられる。

ワープロの普及で，少数言語のテクストであっても印刷用の版下を簡単に作ることができるようになったことは喜ばしいことである。しかし，その反面，印刷用の版下としての役割を果たした後，有効な再利用の機会のないまま，言語学者のハードディスクの中でワープロ文書のまま眠っている貴重な言語資料も少なくないようである。とくに，ワープロ文書は，見た目に正しい文字として印刷され，たいていの人に満足感を与えてしまう点がくせもので，ラテン文字以外の文字や音声記号などの特殊文字を Unicode で処理することが現在の技術でも十分可能であるにもかかわらず，現状では，大部分の言語学者がまだその利用のしかたを知らないでいる。

言語資料を，他の研究者にあげても文字化けしてしまい，活用してもらえない可能性の高い形でせっせと保存している自分たちが，実はたいへん「もったいない」ことをしているのだと気づく研究者たちが増えてくるのをじっと待ち望まなければ

ならないのは，なんとも歯がゆいかぎりだが，本稿では，言語学者たちが，Unicode や XML，マークアップのような最新の言語技術 (language technology, LT) が，自分たちの仕事において果たす役割に気づくのは，もはや時間の問題だという楽観論に立って，現時点で，言語学者たちがこれらの最新技術を活用しようとした場合にぶつかるいくつかの問題を指摘してみたい。

特殊文字体系や音声記号を，最初から Unicode 文字として入力しようとすると，現状ではまだ様々な技術的制約があり，誰でも簡単に始められるというものではない。実際，筆者の所属する東京大学の言語動態学専門分野では，大学院前期課程に入学してくる学生を対象に，毎年，半年間ではあるが，筆者の過去の科研費で開発したツールやフォント([3][4])を実際に導入しつつ，テクスト処理のための Perl 言語によるプログラミングの初歩，テキストエディタの使い方，音声記号などの特殊文字を Unicode で入力するツールの使い方などの手解きをする実習の授業を開講している。

特殊文字や音声記号の多くは，通常，キーボードから直接入力できないから，録音資料を起こして音声表記したまとまったテクストを入力するためには，特殊文字の入力を容易にする補助ツールが欠かせない。たとえば，Windows XP では，キーボードののキー配列を「入力言語」を選んで設定することで，ある程度カスタマイズできるが，「入力言語」のリストに登録されている言語は，世界中の言語のごく一部にすぎないから，少数言語や特殊な文字セットを用いる言語になればなるほど登録されていない可能性が高い。すべての言語ひとつひとつに文字入力のためのキーボードツールを作るよりは，一般ユーザにもカスタマイズしやすいように設計され，十分な汎用性を持つ特殊文字入力用補助ツールの開発は必須である。

なお，これはマークアップ技術の側の問題ではないが，少数言語の場合，個別の言語名コードが国際規格としてまだ決まっていない ([17][18]) ことがあって，言語資料のマークアップの際に，しばしば戸惑う原因となる。たとえば，ウラル諸語の言語名コードを調べてみると図８のようになっている。つまり，国際規格として認

められた言語名コードが，まだ提案中のままになっているものが，ウラル諸語のほぼ半数に達するという現状は，どうひいき目にみてもあまり感心できるものではない。

少数言語になればなるほど，言語コミュニティーにおける有能な言語の使い手や研究者などの人的資源に限界があり，複数の人間による役割分担による効率化が実現しにくく，ひとりの人間が，現地調査によるテクスト収集，テクストの入力，テクストの整形とマークアップまで，家内工業的に，すべての工程に通じていないとことがスムーズに運ばないことが多い。しかし，言語資料が大量に蓄積されてくるにつれ，実際問題として，何らかの役割分担を導入せざるをえなくなることは目に見えている。そのためには，最新の言語技術に通じた人材を少数言語コミュニティからも育成できるようなしくみを整備していくことが望ましい。大学の言語学系の学科や研究機関が積極的に貢献できるとすれば，まず，このような局面においてでであろう。

筆者のこれまでの研究では，音声記号やキリル文字系の特殊文字で表記された言語資料を Unicode で処理するためのフォントや文字入力ツールの開発や，その利用のためのノウハウの蓄積に重点をおいてきた ([1], [2],[3]) 。たとえば，図５のタイプライタと手書きで書かれた音声表記の部分 (ウラル系カレリア語ジョルジャ方言 [13]) は，ウラル言語学で用いられる国際音声字母とは別の音声記号を Unicode 入力し清書したものが図６である。図４のウラル系ベプス語原文とロシア語訳との対訳資料 (約 290 ページ; [12]) は，図７のような形でタグ付けを施す段階まで作業が進んでいる。

2006 年度から新たに始まった新たな科研費プロジェクトでは，一段階進んで，音声記号や特殊な文字で表記された言語資料のマークアップを，文法研究のような狭い意味での言語学的な研究だけでなく，テクスト分析を行う隣接分野の研究でも実際に活用できる形で行うことに重点を移した研究計画をたてている。個々の少数言語について，マークアップされた言語資料を一定量整備し，それを用いて，文法，

語彙，言語変化などの言語学的研究や，テクスト分析の研究を実際に行うことを目指している関係で，具体的にどのようなマークアップ規格を採用するかを決めるにあたっては，実際にコーパスが実現できる方式かどうかを基準にするという現実路線をとることになるだろうと考えている。TEI はマークアップの有力な候補になっているが，XML でマークアップされたテクストを検索するときの技術的問題が，一般にどの程度乗り越えられるものなのかによっても，筆者の研究プロジェクトの今後の展開はかわってくると思われる。

ちなみに，現時点では，XML によるマークアップの階層性を利用したテクスト検索はまだ最先端の技術であって，人文系の研究者，とくに少数言語を研究対象としている研究者たちが気軽に利用できる段階には至っていないのではないか，というのが筆者の受けている一般的な印象である。しかし，大規模コーパスの老舗である BNC コーパス (British National Corpus) が XML マークアップに移行しつつあることを考えると，XML を生かしたコーパス検索ツールが本格的に普及する日は案外早いかもしれない。

5. 最新の言語技術がなぜ少数言語に重要なのか

近い将来，文字言語資料のほとんどが，コンピュータを使って作成された電子的な文書となり，そのままコンピュータを媒介にして交換され，かつ，コンピュータやネットワークに蓄積されて情報検索に利用される時代が到来するものと思われる。そのとき，WWW はもちろんのこと，最新の言語技術 (LT) の恩恵に浴すことのできない少数言語は，残念ながら，現状のままでは，その使用領域が加速度的に縮小していく運命にあるだろうと懸念される。使用人口が少ない言語になればなるほど，言語使用の現場において，電子的に再利用可能な言語資料を産みだすための技術的な制約がますます増大し，その結果，その言語の使用の機会がさらに減少するという悪循環が予想されるからである。もし現実にそうなれば，少数言語を母語とする人々は，これまで以上に，母語のかわりに，多数派の言語で生活することを余儀な

くされ，強力な大言語への同化がますます促進されることになる。しかし，一昔前ならいざ知らず，大言語への同化が進むことを歓迎すべきこととして，成り行きに任せるのをよしとする大言語中心の立場をとり続けるわけにはいかない。

誰も，自分が特定の言語を母語としていることにより，日常生活において不利な状況に置かれることがないように配慮された社会を，言語的にバリアフリーな社会と呼ぶことにしたい。少数言語を母語とする言語的少数者が暮らしやすい社会の方が好ましいと考えるなら，少数言語が社会の隅に追いやられ，次第に消滅していくのではなく，今後も安定して使われ続けることが可能になる条件を社会的に整えて，少数言語を母語とする言語コミュニティを支えてゆく必要がある。

少数言語が今後も健全な形で生き残ることができる前提は，少数言語の立場を考慮した言語技術 (language technology) が確立されることである。言語資料に乏しい少数言語コミュニティでは，現在，新聞発行，出版活動などにおいて，もともと大言語の使用者のニーズに応えるために開発された商業的なワープロソフトやＤＴＰソフトを苦労して使って，自分たちの言語による印刷物を発行するための版下作りを行っている。これでは，他の用途への転用や再活用の可能性が非常に限定された言語資料を次々と産出するために，少数言語の有能な使い手たちの能力が浪費されているといっても言い過ぎではない。少数言語の言語資料も，大言語と同じように，最先端の言語技術を駆使したコンピュータ処理ができるような環境を整えて行くことが望まれる。

6. 展望 — 電子文献学，もしくは，言語資料学

こういった方向に社会が進んで行くためには，伝統的にことばと深く関わってきた大学の文学系・外国語系の学科や，人文社会系の研究機関が，最新の言語技術を使った言語資料の産出・処理のできる人材を育成するための研究や教育に，本格的に取り組む必要があるだろう。この点では，京都大学の人文科学研究所をはじめ，東京外国語大学のアジア・アフリカ言語文化研究所，あるいは大阪の国立民族学博

物館のように，人文系の研究者と情報処理系の研究者が共同研究を組みやすい研究機関の方が，大学の文学系・外国語系の学科と比べ，すでに一歩先んじているかも知れない。

言語の研究は，ヨーロッパの伝統では，もともと philology と呼ばれていたことはよく知られている。Philology を「文献学」と訳したのは，京都大学教授をつとめた詩人の上田敏 (1874-1916) らしいが，「文献学」とならんで「博言学」という訳語も一時期用いられたようである。同じ言語の研究でも，サイエンス，それも自然科学を自称し，「言語学」と訳される 20 世紀の新参者の linguistics と比べると，「文献学」には，図書館の隅の目立たない場所で，誰も関心を持たないような，表紙が変色し，埃をかぶった古い文書を一人孤独にひもといている文系研究者という，あまりあか抜けないイメージがどことなくつきまとう。

広辞苑によれば，「文献学」とは「文献の原典批判・解釈・成立史・出典研究を行う学問」であり，また「それに基づき民族や時代の文化を研究する学問」とされる。よくよく考えてみれば，広辞苑のこの定義は，「文献」を「言語資料」で置き換えれば，電子化された言語資料を出発点として展開されようとする言語の研究のありかたそのままである。とすれば，近頃聞かれる「電子文献学」(computational philology) という形で，言語資料の電子化やそのマークアップに取り組む人文系の研究分野が，この由緒正しい「文献学」という名前を継承するのは理にかなっているように思われる。

ここで注意してほしいのは，「電子・文献学」であって「電子文献・学」ではないことである。従来の「文献学」は，対象が，手稿や印刷物の原典の場合でも，マイクロフィルムなどによる原典の複写されたものである場合でも，人間の目を道具としてテクストを読み，情報を抽出するという方法で行われてきた。「電子・文献学」とは，従来人間の目が担ってきた役割の少なくとも一部をコンピュータに担わせる形で行う「文献学」である。いいかえると，情報抽出やその分析の方法が電子的であるという意味で「電子・文献学」なのである。これに対して，「電子文献・

学」は，従来型の紙のメディア上の文献と対立する意味での電子メディア上の文献，すなわち「電子的な文献」「デジタルな文献」を研究対象とする分野の意味であって，この場合，電子的であるのは研究対象の方である。

出版形態が急速にデジタル・メディアの方向に移行しつつある現在，新しく作られていく文献については，「電子・文献学」でも「電子文献・学」でも，事実上ほとんど違いは出ないかも知れない。しかし，電子メディアによる文献製作が登場する以前の文献を対象とする歴史学や古典学のような従来型の人文系研究分野からみると，「電子・文献学」と「電子文献・学」の間には，天と地ほどの開きがある。電子化された言語資料をマークアップすることで，コンピュータによる文献からの情報抽出を容易にしようとする方向は，まさに「電子・文献学」と呼ばれるにふさわしい。

電子的な言語資料をコンピュータを使って処理することが目新しかった時代ならともかく，それが普通になりつつある時代に，「電子文献学」などと，わざわざ「電子」を冠して呼ぶのはわずらわしいから，単純に「文献学」でいいではないか，という考え方もあるかもしれない。それも一理ある。「文献学」は，たしかに器は古いかも知れないが，新しい酒を盛ってはいけないという決まりはない。今のうちは「電子」とつけて目新しさを誇示してはいるが，いずれは「電子」がとれて「文献学」という名前になる可能性も大いにある。

他方，筆者は，「言語資料学」という名称を数年前から使っている。外国語教育や辞書編纂の基礎と見られがちで，用途が限定されている (言語学的な) コーパスはもちろん，歴史学，民俗学などなど，「ことば」に関わる分野で利用されているテクスト系の言語資料をすべてカバーする概念として「言語資料」(linguistic document) という考え方を前面に出そうという意図である。今後，言語学の下位分野としてのコーパス言語学や談話研究などは，文法研究・語彙研究といった限られた用途のために特別に構築された，狭い意味でのコーパスを使うことから，たとえば，国会の議事録のような言語資料をも，ふつうに研究の対象とするように拡張されていくに

違いない。とすれば，コーパス言語学は，今後，文学研究，歴史学，フォークロア研究，ライフ・ヒストリー研究といった，文字化された言語資料から情報を抽出することが研究の出発点ないし重要な部分になっている人文系の研究分野との共通性をしだいに高めていくことになると思われる。このように考えれば，言語資料を拠り所として展開される研究諸分野を総称する概念として，「言語資料学」を用いてもかまわないのではなかろうか。

名称はともかくとして，言語学の本流の方が，いつのまにか文法研究さえも通り過ぎて，脳科学の方向にどんどん傾斜し，言語資料からどんどん離れている感のある今，言語の研究にとっての原点である言語資料を，ふたたびクールな研究対象として復権することができそうな展望が開けてきたことは，とても喜ばしいことである。テクストのマークアップは，そのためのキーワードのひとつである。

注

〔1〕文法研究や辞書編纂などを目的とする場合，テクストを検索して，ＫＷＩＣ索引などの形で用例を探したり，特定の語についての使用頻度などを調べる必要がある。残念ながら，現在の技術では，デジタル録音された言語資料を直接，自由自在に検索することができないので，何らかの方法で文字転写したものを代わりに用いている。言語の記録という観点からは，文字転写したものは，一次資料としての音声データに対して補助的な役割を果たすにすぎない訳だが，文法研究や辞書編纂の現場では，この関係が逆転し，文字転写し，マークアップしたものの方を主たる言語資料として扱い，一次資料である音声データの方を補助的なものと見ているのが現状である。

〔2〕「危機言語」は，1990 年代の中頃，日本言語学会で endangered languages の日本語訳として採用されたもので，今ではすっかり定着した。ちなみに，中国語では，これを「瀕危語言」と呼ぶようである。

〔3〕音声記号の国際的な標準として知られている「国際音声字母」(International Phonetic

Alphabet, IPA) を提唱した国際音声学会 (International Phonetic Association) の設立は 1886 年。エジソンによるロウ管式蓄音器の発明は 1877 年で，19 世紀末から 20 世紀の初頭には，ロウ管による「珍しい言語」の音声の録音が盛んに行われている。文字を持たない少数言語のテクストが文字化されて，言語学の出版物の中にふつうに現れるようになるのは，おおむね 1910 年代以後である。

参考文献・Web サイト

[1] 松村一登 2002「ロシアのウラル諸語の言語データの収集とその電子化の試み」—『危機言語の現地調査および記述的研究』(平成 11〜12 年度科学研究費補助金基盤研究 (A)・研究成果報告書), pp.51-68

[2] 松村一登 2006「マリ語の言語資料とその電子化」Uralica, Vol. 14 (2006) [印刷中]

[3] 松村一登 2006「ウラル系諸語の言語資料の電子化とマークアップ」—『音声記号等で表記された言語資料のマークアップとコンピュータ処理』(平成 15〜17 年度科学研究費補助金基盤研究 (A)・研究成果報告書), pp.1-30

[4] 鈴木麗爾，小野智香子，松村一登 2003 『フィールド言語学者のための Unicode ツール』(環太平洋の「消滅に瀕した言語」にかんする緊急調査研究成果報告書 B010)

[5] 市河三喜・服部四郎 (共編) 2006『世界言語概説・下巻』研究社

[6] 『国際音声記号ハンドブック』 大修館書店, 2003

[7] デイヴィド・クリスタル 2004『消滅する言語 — 人類の知的遺産をいかに守るか』中公新書

[8] ダニエル・ネトル，スザンヌ・ロメイン 2001『消えゆく言語たち — 失われることば，失われる世界』新曜社

[9] 風間伸次郎 2006『ウデヘ語テクスト２』東京外国語大学アジア・アフリカ言語文化研究所

[10] 静内町文化財調査報告 1995『静内地方の伝承Ⅴ—織田ステノの口承文芸(5)—』静内町郷土史研究会

[11] Nikolaus P. Himmelmann 1998. “Documentary and descriptive linguistics,” in *Linguistics*, Vol.36, No.1: 161-95.

[12] М. Зайцева и М. Муллонен 1969. *Образцы вепсской речи*. Издательство “Наука”, Ленинградское отделение.

[13] Jaan Õispuu 1990. *Djordža karjala tekstid.* Tallinna Pedagoogiline Instituut.

[14] http://www.ethnologue.com (Ethnologue. Languages of the World)

[15] http://www.hrelp.org/documentation (Language Documentation, SOAS)

[16] http://www.tooyoo.l.u-tokyo.ac.jp/ichel/ichel-j.html (危機言語のホームページ)

[17] http://www.loc.gov/standards/iso639-2/langhome.html (Codes for the Representation of Names of Languages)

[18] http://www.sil.org/iso639-3/ (ISO 639 Code Tables, SIL)

図１　母語話者数を基準にした世界の言語の分布 ([14] による)

母語話者数のクラス	現在使われている言語			母語話者数		
	実数	%	累計(%)	実数	%	累計(%)
100,000,000 ～ 999,999,999	8	0.1	0.1	2,301,423,372	40.21	40.21
10,000,000 ～ 99,999,999	75	1.1	1.2	2,246,597,929	39.25	79.46
1,000,000 ～ 9,999,999	264	3.8	5.0	825,681,046	14.43	93.88
100,000 ～ 999,999	892	12.9	17.9	283,651,418	4.96	98.84
10,000 ～ 99,999	1,779	25.7	43.7	58,442,338	1.02	99.86
1,000 ～ 9,999	1,967	28.5	72.1	7,594,224	0.13	99.99
100 ～ 999	1,071	15.5	87.6	457,022	0.01	100.00
10 ～ 99	344	5.0	92.6	13,163	0.00	100.00
1 ～ 9	204	3.0	95.5	698	0.00	100.00
不明	308	4.5	100.0			
計	6,912	100.0		5,723,861,210	100.00	

出典: http://www.ethnologue.com/ethno_docs/distribution.asp?by=size#2 (as of 05-05-2006)

図２　ウデヘ語の言語資料（[9] p.104）

2005年3月14日 クラースヌイ・ヤール村にて録音
N. P. Kukchenko 氏 口述

6. bii guufui-də sagdi samaa bisə
私の 叔父は 偉大な シャーマン だった

77−001
gə xaisi omo bii guufui-də xaisi sagdi samaa bisə.
さあ、もう 一人、私の 叔父（父の姉妹の夫）も やはり 偉大な シャーマン だった。

77−002
guufusini, *znaesh'* sagdi samaa, səwəsiləmi.
亡くなった叔父は、わかるか、偉大な シャーマンで、シャーマンをすると、

77−003
səwəsiləmi sikiə, dəŋə təu puundəgiwənəini bisə.
シャーマンすると、夕方、灯りを 全部 消させるの だった。

77−004
''nii-də əǰiu saulagi. bii jəu-də waami təxəsiǰəŋəi.
「誰も 火を点けるな。私が 何の獣の生贄でも 殺して 皮を剥ぐだろう。

77−005
təxəsiək woosiǰaŋai. təu woosii mətəisini saulagitəuǰə,
剥いで あちこちに投げるだろう。全部 投げ 終わったら 火を点けてくれ、」と、

77−006
gə utə, buji waa-bədə, bujiwə, təxəsii-bədə, təxəsii bisə,
さあ そうして、獣を 殺しているようだ、獣を、皮を剥いでいるようだ、剥ぐの だった、

77−007
uti, dogbo, səwəsimi. mətəə-tənə dianaini, ''gəə, saulagija-ǰa,
彼は、夜に、シャーマンして。終わると、言う、「さあ、灯りを点けろ、」と。

77−008
joktosiami jəu-də anči. təu gaagisii, jauxi, jauxi-ka gaagisiini.
私はよく見たが 何も 無い。全部 持ち去った、どこへ、どこかへ 運び去った。

77−009
tuə-tənə, wakcami, xulimi, baazagatigini tuə, sualaǰami.
冬、 狩りして、歩き回って、タイガへ 冬、スキーで行く。

77−010
ono sualaǰami bimi ono sualai ujələni,
どうやって スキーで行って いるのか、どうやってか 自分のスキーの 上に、

77−011
kəptəmi ŋuai toowa-da əi ila,
横になって 寝る、火さえも 焚かない、

図３　アイヌ語の言語資料（[10] p.364）

	ペッ トゥラシ ウテレケレアン	
	pet turasi uterkere=an	川をさかのぼって走って行った。
	レラメトク アンネプ ネクス	
	re rametok an=ne p ne kusu	私たちは三人の勇者なので
	ホシキノ テレケアンコンノ	
	hoski no terke=an konno	先に私が走ると
	イオシ アコロ オッカイポウタラ	
2040	i=os a=kor okkaypo utar	私のあとから私の若い者たちも
	イオシ	
	i=os	私のあとから
	イノシパ フミ ネコトム イラムアン	
	i=nospa humi ne kotom iramu=an wen …	追いかけてくる音がするように思えた。
	ウェン…ケウトゥム ウェンカ アキ	
	wen … kewtumu wen ka a=ki	私は思いが悪くもなり
	ウェン イルシカ アキ	
	wen iruska a=ki	ひどく怒りもした。
	ウェンカムイウタラ エネ ヘンネ キヤクン	
2045	wenkamuy utar ene henne ki yakun	悪者たちがあんなふうにしなかったら
	アンロンノカ ソモ キ	
	an=ronno ka somo ki	私は殺しもしない
	アンチセウコウフイカカ ソモ キイケ	
	an=ciseukouhuyka ka somo ki hike	家といっしょに燃やしもしなかったのに
	ウェイサンペ コロワ	
	wen sampe kor wa	悪い心がけを持って
	アコロ オナ コロペ オピッタ ルラクス	
	a=kor ona kor pe opitta rura kusu	父さんのものを全部持ち去ったので
	ウェンカムイウタラ ポクナモシリ アコキル	
2050	wenkamuy utar poknamosir a=kokiru	悪者どもを地獄に突き落としたのだ。
	アコシユプ カムイ ヌプリ	
	a=kosiyupu kamuy nupuri	私は気合いを入れて神の山
	ヌプリ トゥラシ スイ	
	nupuri turasi suy	山を登ってまた
	テレケアニネ リキプアニネ	
	terke=an hine rikip=an hine	走って登って
	アコロ オナ コロペウタラ	
	a=kor ona kor pe utar	父さんの持っていたものを
	スイ アンウサライェイネ	
2055	suy an=usaraye hine	また分け合って
	アイセ エアシカイ パクノ	
	an=se easkay pakno	運べるだけ

図４　ベプス語の言語資料（[12] p.3637）

Иняков Василий, 15 л.
Маг. зап. 166/3. М. Муллонен, 1961.

16. príha ostí bazarou źerklon

elí endę ükś príha. ťedan, mäń hän bazaralę i ostí śigä źerklon. kodhe tuĺ, kaclęb źerklohо: «čoma olęn». ak ńecęn homęič, dumę,ib: «minak hän śinna kaclęb?». konz hän ĺäkś (príha) kodišpęi, ak kacob: «a, sanob, ńügüde ťedan, keda kaclęb. hän, sanob, bazarou ĺüuźi ńeičen da sęn karťinąine om. tulęške, sanob, mamoi, kaco». mamazę tuĺ, kacuhťi: «ka, sanob, om babka ńečiť, minun vuiťťe». śit tuĺi tatazę: «ka min ťii paǵižętęi, om ńečiť mužik, bardanke da furaškoiš». śid vähäinę da tuĺ iče hän príha. he ǩüzęltasęi: «kedak śina bazarou ĺüuźid?». — «a kedak om?». — «ka mii naku kacuimęi, sanotas, da em ťea, ǩen om». a hän śiižuťi heiť kaikid ühtęs da ozuťi: «vot, sanob, ťii iče olętęǵi».

16. Парень купил на базаре зеркало

Жил раньше один парень. Пошел он на базар и купил там зеркало. Пришел домой, посматривает в зеркало: «Красивый я». Жена заметила это, думает: «Что он туда посматривает?». Когда он ушел из дому, жена смотрит: «А, говорит, теперь знаю, кого рассматривает. Он, говорит, нашел на базаре девушку, да это ее карточка. Иди-ка, говорит, мама, посмотри». Ее мать пришла, посмотрела: «Да это, говорит, бабка, похожая на меня». Потом пришел ее отец: «Да что вы говорите, это мужик, с бородой да в фуражке». Потом через некоторое время пришел сам парень. Они спрашивают: «Кого ты на базаре нашел?». — «А кого?». — «Да мы тут смотрели, говорят, да не знаем, кто». А он поставил их всех вместе и показал: «Вот, говорит, сами вы и есть».

図５　カレリア語の言語資料（[13] p.175）

98. / s/l e p u v u t t i h š i l m ä t /

s/lepuvuttih //

/ k o ž ? /

jo ka počti• vuaž hänel // vuaž // üks silmäst čirkzen nägöw / a toin / toin ei näw // i / i apera•cid ei ruvet ruadmah // näil oll/ah omat / m o s k u š //i to ei ottuče // mož fofse• rikot // razv čirkzen nägüw // ain šanow // "kuin vihmuw / fos-sendah nakroičet / vrod on sol/nj/škan" // tüt ei niä // ei // ühel šilmäl // üksin el i / hänen kodin on atale•nnest kaik küläš // a heidäh i / i kaik siel / on seičmen kodi / i koih müt-ten oška / šin vanhat // nu händ ei kačot / što hiän on / al/e/loš // šid mändih / a hiän kua/liä //

ML Sem 85

/ Silmad jäid pimedaks? /

Jäid pimedaks.

/ Millal? /

Juba peaaegu aasta tagasi. Aasta. Ühe silmaga natuke näeb, aga teine, teine ei näe. Ja, ja operatsiooni ei hakata tegema. Neil on sugulased Moskvas ja needki ei võta (operatsioonile). Võib-olla rikud täitsa ära. Siiski natuke näeb. Ütleb alati: "Justkui sajaks, justnagu oleks päike." Tüdruk ei näe. Ei, on ühe silmaga. Üksi elas ja ta majake on külast eemal. Aga neid on... Seal on seitse maja ja kõik (elanikud on) vanad. Tema järele ei vaadatud, on ta elus või ei. Siis mindi, aga ta on surnud.

図６

ウラル音声表記 (UPA)のテクスト(図５)を Unicode 対応フォントで清書したもの

/ sлepuvuttj̮h šil´mät /

sлepuvuttj̮h //

jo ka počti· vuaž hänel̄´ // vuaž // ükś šil´mäšt čirkzen ńägöw / a toiń / toiń ei ńäw // i / i apera·cid ei ruvet ruadmah // ńäil̀´ oллah omat / moskuš // i to ei ottuče // mōž fofśe·rikot // ràzv čirkzen ńägüw // ain šanow // « kuin vihmuw / fośśendah nakroičet / vròd́ on soлnj̮škań » // t́üt̀´ ei ńiä // ei // ühel̄´ šil´mäl̄´ // ükśin el´j̮ / hänen kod́in on atal´e·nneśt´ kaik̀ kül´äš // a heidäh i / i kaik̀ šiäl̀´ / on seičmen kod́ī / i koih müt´t´eń oška / šin vanhat // nu händ ei kačot /što hiän on / al´ eлoš // šid mändih / a hiän kuaлiä //

図 7

ウラル音声記号で転写されたベプス語のテクスト(図４)に最小限のタグ付けを試みたもの

```
<div3 id="16">
<informant>Иняков Василий, 15 л.</informant>
<doc_info>Маг. зап. 166/3. М. Муллонен, 1961.</doc_info>
<div4 id="16_1" lang="vep">
<h>16. príha ostʹi bazarou źerклon</h>
<p>
<s no="0671"> eĺi endę ükś príha. </s>
<s no="0672"> tʹedan, mäń hän bazaralę i ostʹi śigä źerклon. </s>
<s no="0673"> koďhe tuĺ, kaclęb źerклoho: «čoma olęn». </s>
<s no="0674"> ak ńecęn homęič, dumę,ib: «minak hän śinna kaclęb?». </s>
<s no="0675"> konz hän ĺäkś (príha) koďišpęi, ak kacob: «a, sanob, ńügüde tʹedan, keda kaclęb. </s>
<s no="0676"> hän, sanob, bazarou ĺüuźi ńeičen da sęn kartʹinąine om. </s>
<s no="0677"> tulęške, sanob, mamoi, kaco». </s>
<s no="0678"> mamazę tuĺ, kacuhtʹi: «ka, sanob, om babka ńečitʹ, minun vuitʹtʹe». </s>
<s no="0679"> śittuĺi tatazę: «ka min tʹii paǵižętęi, om ńečitʹ mužįk, bardaŋke da furaškoiš». </s>
<s no="0680"> śid vähäine da tuĺ iče hän príha. </s>
<s no="0681"> he ĺküzęltasęi: «kedak śina bazarou ĺüuźid?». </s>
<s no="0682"> – «a kedak om?». </s>
<s no="0683"> – «ka mii naku kacuimęi, sanotas, da em tʹea, ĺken om». </s>
<s no="0684"> a hän śiižutʹi heitʹ kaikid ühtęs da ozutʹi: «vot, sanob, tʹii iče olętęǵi».
</s>
</p>
</div4>
</div3>
```

図８　ウラル諸語の言語名コード ([17][18] による)

		ISO 639-2 & 639-1	ISO/DIS 639-3
フィン・ウゴル語(その他の)	Finno-Ugrian (Other)	fiu	
エストニア語	Estonian	est (et)	est
フィンランド語	Finnish	fin (fi)	fin
メアンキエリ語	Meänkieli		fit*
カレリア語	Karelian	krl	krl
オロネツ語	Livvi [Olonetsian]		olo*
ベプス語	Vepsian		vep*
イジョール語	Ingrian [Izhorian]		izh*
ボート語	Votic [Votian]	vot	vot
リーブ語	Liv [Livonian]		liv*
サーミ語(その他の)	Sami languages (Other)	smi	
イナリ・サーミ語	Inari Sami	smn	smn
ルレ・サーミ語	Lule Sami	smj	smj
北サーミ語	Northern Sami	sme (se)	sme
スコルト・サーミ語	Skolt Sami	sms	sms
南サーミ語	Southern Sami	sma	sma
アカラ・サーミ語	Akkala Sami		sia*
ケミ・サーミ語	Kemi Sami		sjk*
キルディン・サーミ語	Kildin Sami		sjd*
ピーテ・サーミ語	Pite Sami		sje*
テル・サーミ語	Ter Sami		sjt*
ウーメ・サーミ語	Ume Sami		sju*
マリ語	Mari	chm	chm
東マリ語	Eastern Mari		mhr*
西マリ語	Western Mari		mrj*
エルジャ語	Erzya	myv	myv
モクシャ語	Moksha	mdf	mdf
コミ語	Komi	kom (kv)	kom
ウドムルト語	Udmurt	udm	udm
ハンガリー語	Hungarian	hun (hu)	hun
ハンティ語	Khanty		kca*
マンシ語	Mansi		mns*
セリクプ語	Selkup	sel	sel
ネネツ語	Nenets		nen*
ガナサン語	Nganasan		nio*

[注] * はまだ国際規格として認められていない(提案中)のもの

マークアップの課題を syntax から見た分類と解決のステップ

大矢一志
鶴見大学

人文科学研究で使われる資料を電子化し、それにマークアップ (markup) を施す際に困難を感じる原因は、主に 3 つ、1) 対象データの分析が十分でない、2) マークアップという手段が本来の目的と合わない、3) マークアップ技術の理解が十分ではない、ことが考えられる。但し、3) にあるマークアップ技術は、まだ十分に成熟したものではなく、そのため、例えば XML といった規格自体が持つ不備が原因で「上手く書けない」ことがある。特に、XML は、アプリケーション (e.g. TEI もそのひとつ) を複数関連づけることが規格上困難であるにも関わらず、多くのアプリケーションが提案され、利用されている。実は、複数の XML アプリケーションを関連づけ、統合する方法は、ML(markup languages) の専門家でも解決策は一意に定まらない。単なるデータ入力や変換をするのではなく、はじめからどう書く (マークアップす) べきかを決めることは、かなり高度な作業になっている。

しかし、マークアップすること自体は、人文科学研究者にとってはとても身近な行為である。本稿では、マークアップする際に困難と感じる原因のうち、ML の Syntax から見た「ひっかかりどころ」を紹介し、規格の不備に惑わされることなく、マークアップが本来持つ自由な記述を再確認したい。これは、TEI を利用する際、「ML 一般」と「個別テキストタイプ」という 2 つの問題を扱う TEI の論議を、整理して読み進めるヒントとして有効だろう。さらに、ML 一般の問題を検討する際の、手助けになるかもしれない。ML は、単に利用される規格としてあるだけではなく、従来、ひとがアノテーションとしてきた記述の行為が、形式言語の側面からメタ記述の行為として評価されうる可能性を探る糸口にもなっている。

Markup problems:Syntactical analysis and steps to their resolution

OHYA Kazushi
Tsurumi University

In the process of digitalizing textual ressources used in the humanities and subsequently adding markup to them, there are, I think, mainly three reasons that difficulties are encountered: 1)the source material does not have been sufficiently analyzed, 2)markup as a method is at odds with the original aims, 3)markup technologies as such have not been sufficuently mastered.

However, the last point also includes the fact that markup technologies themselves have not yet matured sufficiently, so that because of the way the XML standard is defined, some needed constructs can not be written easily. Among other things for example, although the combination of several XML applications (among them for example TEI) is difficult to achieve because of the way the standard is defined, there have nevertheless a large number of applications been defined and widely used. This combination of several applications is something even specialists in markup languages achieve only with difficulties. Not just data input or data conversion, but decide how to encode what is indeed a task that requires high level skills.

On the other hand, adding markup itself is something that is very close to home for a scholer trained in the humanities. In this paper, I will focus on one of the above difficulties, that is the "syntactically induced" problems and pitfalls of markup languages. In the application of TEI, some problems encountered are due to the way markup languages as such are defined, others result from the specific text type used. This differentiation is helpful in understanding the way such problems are handled within the TEI, but they can also be applied to markup languages in general. Markup languages are not something that should simply be used since it is defined as a standard, but since they use formal languages to apply annotations as a description on a meta-level, they provide a means to analyze and reflect upon this act as such.

1　はじめに

本稿の目的は、マークアップ言語を使った資料作成の手法が、日本の人文科学研究でまだ一般の手法とはいえない現状の原因として考えられる、現行マークアップ言語によるアノテーションに伴う技術的な困難さを整理・明示することにある。この技術的な困難さは、マークアップ言語の専門家では共有されている感覚ではあるが、1) 普及を目標とした活動で負の点に言及したくないという政治的な判断、2) 技術的問題を解決する知識の提供を生業とする場合には言及しづらいという営業上の判断、3) 技術の話をする人文科学研究の会合が少ないという機会不足、4) 専門家にとっても難しい問題自体の複雑さ、そして 5) この問題を検討すること自体が、技術者にとってはあまり有益ではないという価値の不明瞭さ[1]、といった理由から、明文化されることが殆どない。また、利用者は、マークアップ言語に、プログラミング言語と同程度の困難さを感じ、マークアップ言語を使った資料作成の困難さの原因を、自らの知識不足と感じる傾向がある。知らされていない技術的課題が、不足している知識とされ、結果、検討する機会が失われている。

本稿では、マークアップ言語が抱える課題のうち、1) アノテーションとして使われる際に、2) マークアップ言語の Syntax 上で感じる、3) 技術的な課題を整理し、人文科学研究でマークアップ言語を利用した資料作成の活動を始める際の検討材料を提供したい。

本稿は、以下の構成になっている。まず、マークアップ言語の技術的な課題を紹介する前に、マークアップ活動を始める前提について確認をしておきたい。次に、アノテーションを行う際に、現行マークアップ言語に感じる不都合さを紹介し、その原因を解説する。次に、アノテーションを目的としたマークアップ言語一般の論議を整理する。最後に、ここまでの観察から導かれる、TEI を使うメリットを紹介し、同時に、日本で TEI を利用する際のスタンスを確認したい。

2　マークアップ活動の前提

マークアップ言語を使ったアノテーションの技術的課題を確認する前に、マークアップ活動を始める前提を確認しておきたい。

XML の普及によりマークアップ言語を使ったデータ作成が、一般的な手法として広く知られるようになってきたが、時に、マークアップ言語の使用が、当該プロジェクトの最終目的にとって適切であるのか疑問であるケースが見られる。マークアップとは、いわゆるテキストデータに対して、マークアップ言語を使用して行われるアノテーション行為のことである。この行為の対象は、電子化されてあるテキストで、これには、生来デジタル (born-digital) なものと、デジタル化 (digitized) されたものがある。人文科学系の資料の場合、テキストデータの主流は、デジタル化されたテキストである。デジタル化される場合、その結果としてあるものは、テキストデータである必要はない。また、必ずしもマークアップされる必要もない。例えば、プロジェクトの目標が、動画との連携してテキストを表示することであれば、アニメーションとして動く文字の方が、潜在的に情報 (意味) を伝達する視覚表現として高い効果を発揮するだろう。一般に、デジタル化は、表現力を高めるために行われると考えられる。例えば、着色、立体化、卓上化 (縮小化)、マルチメディア化などは、元媒体にある制約を超えた表現力を得るための行為である。一方、いわゆるテキストデータは、デジタル化されたテキストではあるが、例えば、符号化文字集合や線的であるという性質が、いわゆる媒体の持つ制約条件としてあり、表現力に一定の制限が加えられている。つまり、表現力を高めるためのデジタル化であれば、結果としてあるものがテキストデータである必要はなく、さらに、それをマークアップする必要もない。XML の場合、データラッパー[2]としての利用も想定されているのであるから、どのような形であれ、XML 形式でデータが出力できれば、十分である。マークアップされたテキスト

[1] 哲学的な論議と感じてしまい、敬遠される傾向がある。結論を先に言えば、後述するよう、マークアップ言語の論議には、人文科学的なスタンスも必要になるため、この直感は正しい。

[2] システム間でデータを交換する際に採られる手法の一種で、共有 (標準) 符号化方式といったもの。テキストデータであることを前提とすれば、共有デリミタともいえる。

を中心とするのであれば、そのプロジェクトは、テキストデータが持つ、1) 人が読めること、2) 内容の保守が容易であること、3) データ寿命が延びる可能性が高まること、ひいては 4) 再利用性が高まること、5) 保存性が高まること、そして 6) ユニバーサルな検索対象形式であること、などの利点を生かしたデジタル化の目標を持つものになる。表現力を高めるプロジェクトでマークアップ言語を使い、マークアップの手法に否定的な考察を導くのは、正当ではないだろう。もちろん、マークアップはデジタル化の手法でもあるから、排他的な関係ではない。この包含関係を理解し、適切な手法を採ることが、マークアップの問題解決になる場合がある。

マークアップ活動を始める前にもうひとつ確認しておきたい前提は、記述対象となる元資料の分析が十分に行われているかである。この当然のような確認は、マークアップ言語の利用者に向けて、知的怠慢を牽制するといった警告ではなく、むしろ、現行マークアップ言語が抱える問題への対処法として、利用者へ自己防衛を促すための警告のようなものである。マークアップ言語について常識程度の知識は必要であるが、敢えていえば、アノテーションの際に、現行マークアップ言語が難しくし感じる点の多くは、現行マークアップ言語にある不備が原因としてある。現行マークアップ言語は、データラッパーとしての利用を中心として技術的な検討が行われ、アノテーションとしての使用は十分に想定されてこなかった。結果として、アノテーションに不都合な技術的な課題が多く残されている。事前に資料分析を十分に行うことは、マークアップが困難である原因を上手く振り分けるという、規格の不備に惑わされない対応策となる。

本来であれば、マークアップ活動は、マークアップすることにより構造が見えるという、書くこと本来の活動と同じ生産活動であるから、書きながらの分析というのは、至極自然なアプローチである。ところが、現行マークアップ言語は、そのような (自己) 発見行為に十分応えることができない。アノテーションにマークアップ言語を上手く利用できないことは、実は、マークアップ言語の本質的な課題と関連している。この問題は、幅広い論議が必要となるため、詳細は別の機会に譲り、本稿では扱わない。

3 現行マークアップ言語

3.1 利用の現状

日本の人文科学分野において、TEI に限らず、マークアップ言語による資料作成は、メタデータを対象にした活動は盛んであるが、本文を対象とした活動となると、あまり報告を聞かない。理由は様々であろうが、そのひとつとして、現行マークアップ言語に対する素朴な実感が原因としてあるだろう。

XML は、データ交換の共通マークアップ方式として規定され、具体的な語彙までは規定していない。XML アプリケーションとは、特定分野で共有されることを目的に作られた語彙集合のことである。現在、多様な XML アプリケーションが提案されている[3]。実際に XML を使った資料作成を計画すると、まず XML アプリケーションの多さに戸惑うことになる。そして、現状では、それらの選択基準が存在していない。資料の分析が十分に行われていても、それに叶うアプリケーションを分析する手法がないのである。従って、XML アプリケーションの選択は恣意的になり、ここに、特に研究者は、抵抗感を感じることになる[4]。

また、XML アプリケーションを検討し、例えば、複数の XML アプリケーションの使用が望ましいと判断された場合でも、複数の XML アプリケーションをどう使用するかで、戸惑うことになる。現行では、複数の XML アプリケーションを同時に使用する手法が決められていない。その調整は利用者に任されている。これは、現実にはほぼ不可能といってよい[5]。

既存の XML アプリケーションを使用するのではなく、独自開発を選択した場合には、マークアップ言語に関する相当の知識と、記述対象資料への深い分析が求められてくる。これには、相当の時間と労力が求め

[3] cf. `http://xml.coverpages.org/`
[4] これは、SGML の時代からの課題である。
[5] 名前空間 (Namespaces in XML) による解決が、一時期、検討されていたが、現在では、これによる構造の調整はできない。

られるため、プロジェクトの目的が、マークアップ言語や符号化方式の開発自体を目的とするのではなく、人文科学資料の一層の利用を促進するものであれば、この選択は難しい。

結果として、資料にマークアップを施す作業を実行に移すことは、極めて困難になっている。

3.2 規定・論議の状況

現行マークアップ言語は、SGML の流れを汲む XML がデファクトスタンダードになっている。マークアップ言語には、SGML 系以外にも、様々なものが提案され、実用されてきた。例えば、初期のマークアップ言語としては、roff 系、電算写植系、TEX 系等が広く利用されている。これらには、主に割り付け情報やその単位と処理を指定する為に使用されてきたという共通の特徴がある。SGML の開発時期は、これらと同時期のものであるが[6]、現在では、マークアップ言語の基本形であるような扱いをされている。残念なことに、この間、マークアップ言語一般についての論議は、あまりされていない。例えば、そのような論議を進めてきた数少ない団体である TEI においても、SGML のアノテーション能力や、それに相応しい SGML 機能の選択などの論議は行われてきたが、SGML を形式言語として扱う分析は殆ど無い。SGML は、マークアップ言語の基本的な論議がないまま、現行 XML までの主流を形成してきたといえる[7]。

ところが、実用から観察されたものでもなく、自然から発生したものではなく、理論から導かれたものでもない、SGML 系マークアップ言語について、現在では、様々な分野からの分析が行われている。例えば、DB の視点から探る S.Abiteboul et.al.2000[1] では、マークアップ付データを semi-structure data と表現し、SGML/XML 構造にある情報をどのように表形式へ効率よく確実に書き換えるかについての研究が行われている。現在では、SGML/XML 構造にある情報をそのまま扱う DB の開発が行われている。また、パーサの立場から探る H.Hosoya and B.C.Pierce2000[6] では、効率よく SGML/XML 構造を解析するオートマトンの発見と、同時に、SGML/XML 構造の形式表現 (形式言語) について研究が行われている。また、オートマトンの立場から探る F.Neven et.al.2004[10] では、マークアップ言語によるアノテーションが行われた結果としてある文字列の内、メタ記述部分と地の文とを、同時に扱うオートマトンの研究が行われている。これらの研究によって、マークアップ言語は、単なる規格の対象ではなく、新しい研究領域の対象とされた。とりわけ、アノテーションとしてのマークアップ言語研究において、新しい形式表現を探る研究が行われたことは、マークアップ言語の根本的な論議を進める下地にもなることが期待できる。但し、これらの研究の土台となるマークアップ言語そのものの意義、すなわち、アノテーションの結果として生まれるメタ記述を表記するためのマークアップ言語として何が求められているのかが漠然としたままでは、現行規格の応用状況を説明するだけの研究に留まる。我々が必要としているのは、アノテーションとして相応しいマークアップ言語であり、そのための理論や具体的な文法の提案である。そのためには、アノテーション行為で必要とするメタ記述とそのメタ言語は、記述実験を通して十分に検討される必要がある[8]。

3.3 アノテーション時に感じる不満

アノテーションの道具として、現行マークアップ言語を使用する際、その Syntax 定義に不便さを感じることがある。例えば、思いついた時に、好きなところにアノテーションを行うことが、現行マークアップ言語ではできない。その前に、根要素 (a root element) を選択・記述する必要があり、場合によれば、その構造定義を宣言する必要がある。また、マークアップの際には、要素名を決める必要があり、更に多くの場合、当該マークアップ部が、全体の構造の中でどのように位置づけられるのかを決める必要がある。つまり、現行マークアップ言語は、アノテーション行為に先立ち、その内容が既決であることを前提に使用されるつくりになっている。

[6] 1967 年開発、1986 年 ISO 化。
[7] 結果として、マークアップ言語の論議が、規格の論議としか見なされないという文化も生まれた。
[8] メタ言語で、書ける・書けないの論議が、もっと活発に行える環境が必要である。

3.3.1 根要素

マークアップの際に、根要素が不要であれば、アノテーションは書きやすい。例えば、XML データではない素テキスト (plain text) データに、注釈としてタグ付けしてあるものも、現行 XML データと同様に扱えれば、便利である。同様のことは、ラッパーとして利用される場合でも見られる。例えば、Java に見られる、あらゆる設定を XML で記述する流れの反動として、Rails[5] では、テキストデータによる設定が提案され、歓迎されている。マークアップとは、自由記述に制限を課し、処理を可能にした技術のひとつであるから、記述が冗長となるのはマークアップの本質で、仕方がない。しかし、本質に関わる冗長性なのかは、十分に検討する必要がある。例えば、根要素も、その候補である。SGML/XML では、根要素を必須としているが、これは、構造に関わる 1) 構造定義、2) 構文解析、3) インスタンス指定の 3 つの場面何れにおいて、無くてもよい。構造定義において、根要素は、当該データの構造名を示し、構造定義ファイルとの関連を取る ID の役割を果たしている。しかし、この役割は根要素である必要はない。例えば、ハブデータを介することで、構造名を当該データ中に埋め込む必要は無くなる。構文解析において、根要素は、解析対象文字列の外枠を示しているが、現行規格ではファイル全体が解析対象であり、根要素はその領域を規定する働きをしていない。well-formedness の制約があれば、複数の木要素の解析は問題にならない。反対に、根要素を必須とせず、代わりにデフォルトの仮想根要素を想定すれば、素テキストもマークアップ付データとして扱うことが可能である。インスタンス指定において、根要素は、path を使った指定で、その先頭要素として使用されている。木構造上の path はユニークであるが、根要素はその弁別素性として働いていないため、無くとも問題ない。

但し、根要素を選択的なものとして、例えばデフォルトの根要素を想定した場合、従来、扱いが曖昧であった無名内容 (pelement[8])[9]を積極的に使用する必要がある。無名内容とは、混在内容 (mixed-content) 中にあるテキストデータ部分である。現行規格では、構造定義やインスタンス指定において、無名内容は統一的な扱いがされていない。従って、例えば、無名内容と兄弟要素間との連携を明記することは難しい。この背景には、マークアップ部分とその内容部分 (いわゆる地のテキスト部分) とを同時に扱う理論がまだ十分に検討されていないという理由がある。もし、無名内容を積極的に扱うことができれば、アノテーションの記述技法は、より豊かになることが想定される [12]。デフォルト根要素や無名内容の扱いは、アノテーション向けのマークアップ言語を検討する際の重要な課題である。

3.3.2 要素、構造、名前

現行マークアップ言語において、要素の名前と構造は、一体化している[10]。要素名を使用する際には、当該要素の構造が確定されている必要がある。また、要素の構造情報だけを構造指定、解析対象、インスタンス指定で使用することはできない。一方、アノテーション行為には、必ずしも事前の分析・検討が十分であるという前提はない。むしろ、人文科学研究においては、分析を進める行為がアノテーション行為である場合が多い。従って、a) 構造が既定である要素しか使えないこと、b) 要素名とその構造が一体化していることによって、1) 名前に伴う構造を変更できない、2) 同じ要素名で構造が異なることを指定できない、3) 異なる構造を同じ要素名で扱えないことは、アノテーション行為にとっては不便である。

しかし、データ交換形式として XML を使用する場合には、インスタンスのデータ単位が指定できればよく、この場合、当該インスタンスの構造は既知であるから、要素名によってデータ単位を指定することができる。つまり、データラッパとしてのマークアップ言語にとっては、要素の名前と構造が一体化していても問題がない。

[9]従来は「疑似要素」と訳されている。先日の発表では「無名要素」としたが、本稿では、無名内容 (obscure/anonymous content) と無名要素 (obscure/anonymous element) とを分けて論議を行いたい為、この名称に変更した。

[10]XML の規定にあるような"a type of an element" では決してない。"a name of an element" でしかない。

本来、マークアップ言語一般の論議としては、要素の名前と構造を分けて扱う可能性がある。また、無名内容のように、構造特性のみによる情報単位の扱いが可能になれば、無名の要素も想定した仕組みが考えられる。さらに、無名要素の導入に伴い、開始タグと終了タグのペアは、要素名で指定される必要がなくなるため、個別指定を行うことも考えられてくる。これらの考えは、従来にない、全く新しい仕組みのマークアップ言語を生み出す可能性を秘めている[11]。但し、現時点では、その有効性は不明である。現状では、現行マークアップ言語で記述されるデータ構造は、線形の文字列上に作られる木構造である。しかし、アノテーションで示される情報がそれで十分に表現できるのか、実は、確証がない。例えば、同時性やオーバーラップなどは、木構造上に上手く表現できない情報種として、以前から論議されてきた。アノテーション行為において必要なマークアップ言語の記述力とは何かを明確に挙げられる程、十分な記述実験が行われていないのが現状である。

以上から、要素の構造は、必要な情報を十分に表現できる保証がなく、またそれが可能だとしても、例えば、インスタンス指定の際には不要であることが判る。この現状では、素朴に、マークアップ言語を使うアノテーションでは、構造は不要であるかの印象を持つことになる。ひどい帰結ではあるが、これが現行マークアップ言語が抱える、根本的な課題である[12]。SGML の時代から、マークアップ言語の論議では、文書構造の記述が、マークアップ言語の本質として主張されてきた。論理的文書構造の存在とその重要性を、もはや現時点で否定することはできないだろうが、現行マークアップ言語がその記述に十分であるのか、とりわけ、アノテーション向けのマークアップ言語として、論理要素を表現するに十分な機能を備えているのかを検討することは、今後の課題である[13]。

4　アノテーションと現行マークアップ言語

アノテーション指向マークアップ言語というものが存在しない現状では、現行マークアップ言語を使い、アノテーション行為を行わなくてはならない。その際、マークアップ言語へのスタンスは、1) スキーム[14]を独自に開発するか、2) 特定 (XML) アプリケーションを使用するか、によって、大きく異なってくる。

4.1　スキームの独自開発

スキームを独自に開発する場合、現行マークアップ言語の癖に対応して、構造と要素名の捉え方には、以下のスタンスが考えられる。

4.1.1　構造の考え方

構造が判らないでいるデータに、構造を付ける必要はない

現行マークアップ言語では、1) 同じ要素名で異なる構造の指定は困難、2) 線形文字列上に作られる木構造しかない、3) インスタンス指定時には構造は不要、そして 4) 必要な抽象構造なのかを書きながら検討できない。抽象単位である非終端データ単位を無理に想定する必要は、全くない。構造は上向きに検討するもので、例え flat な構造でも、インスタンス指定時には要素名の工夫により十分に対応できる。構造は無理に作るものではない。

[11] 現在、これらの特性を持った新しいマークアップ言語を検討している。アノテーションに関わる多くの要求に対応できるものにしたい。

[12] なぜ技術者がこれを語らないか、これでよくお分かり頂けると思う。

[13] 単なる規格としての論議を超える研究という位置づけの他、アノテーション行為を外延的に定義しうる新しい研究領域の開発という意味でも、重要であると考えている [2]。

[14] 文書 (要素) 構造のこと。以前は、scheme と表記されていたが、近年は、schema(schemas, schemata) と表記されることが多い。本稿では、XML Schema との混乱を避けるために、昔風の scheme を採用した。

根要素は無視しても構わない

現行マークアップ言語では、根要素は必須であり、構文解析等、支援ソフトを使用する際には必要になる。但し、ハブデータを使用すれば、コンテンツを含むデータ (ファイル) に根要素は必要ない。アノテーション行為にかける時間と、パーサにかける時間とを比較すれば、根要素を使わない選択が考えられる。

構造定義は気にしない

アノテーション行為において、構造定義は、アプリオリにあるものの他、インスタンス作成後に、結果としての構造定義をアポステオリに得るという使い方も考えられる。アノテーション行為の本質からしても、構造定義に拘る必要はない。また、その他にも、1) 構造定義についてアプローチが統一されていない、2) 使用するソフトが制限される、3)(現行では) 排他的にしか構造定義は使用できない、4) 必要な指定ができない、5) 無くてもよい (well-formed format が認められている)、という点から、構造定義を無理に求めることはない。

4.1.2 要素名の考え方

属性を弁別素性として使用する

現行マークアップ言語では、インスタンス指定方法として、1) 要素名、2) パス、3) 順番 (兄弟要素間) が利用されている。中でも要素名は、構造名であるだけではなく、インスタンス指定の際、参照情報として中心的な役割を果たしている。考え方として、もし要素名をユニーク化した場合、パス名を使う必要はなくなる。パス名は、順序付要素名列であるから、その構成要素がユニークであれば、順序付要素名列全体で指定する必要はなくなる。考え方として、もし要素名を一般化・単純化した場合、パス名は、要素名列を弁別素性にしたものとして使用できる。更に、属性を弁別素性として使用した場合、属性付要素名は、個々のインスタンス単位 (ノード) までを指定する弁別方法として使用できる。また、属性は、パスとは異なり、構造に直接関連する順番とは無縁の、集合要素として弁別素性とすることができる。アノテーションにとって馴染みやすい手法は、後者であろう。一般化・単純化された要素名に、属性を弁別素性として使用することで、アノテーションに相応しい柔軟な記述が可能になる[15]。

4.2 既存 (XML) アプリケーションの採用

スキームを独自開発するのではなく、既存の XML アプリケーションを採用するのであれば、以下のスタンスが考えられる。

XML アプリケーションは、複数使わない

現在、複数の XML アプリケーションを調整する規格・規定はない。この調整は、専門家においても答えが一意に定まらないのが現状である[16]。従って、一般には、構造定義類も含めて、複数の XML アプリケーションを同時に使わない方がよい。特に、長期間の利用が想定される人文科学資料をマークアップする際には、XML アプリケーションを複数選択する利点を、慎重に検討すべきである。

[15] 属性、要素名、要素内容の使い分けについては、従来から様々な論議が混沌としてある。その多くは、メタ言語一般の論議としてされているが、本稿のは、現行規格から考えられる、使い分けのひとつの提案として検討している。但し、この論議は、規格の論議に陥る、典型的な例であると考えているので、TEI の論議を促す会議での発表としては相応しくなかったと、反省している。

[16] メタ言語のインスタンスによるメタ言語の規定、というメタメタ定義についての論議は、実装側から嫌われ、結果として XML が生み出されたが、XML アプリケーションの関連性を規定すること (メタ性の定義) は、根本的な問題ではなかったのかと、今となって改めて思う。現世代が責任を持つべき重要課題であろう。

アプリケーションの選択では、派生規格の有無から考えられる将来性ではなく、ドキュメントやソフトウェアの充実度を検討する

XML アプリケーションは、モジュールとして複数同時に使用することが困難であるから、その選択は慎重に行う必要がある。XML アプリケーションは、規格として、将来の利用を前提に提案されているが、それが現実となるケースは、希である。検討の際には、規格としての将来ではなく、現時点での利用環境の充実度を基準に検討を行うべきである。

構造定義類のアプリケーションは、無視してもよい

構造定義類は、重要規格とされているが、XML アプリケーションである。また、アノテーション行為にとっては、先述したような理由から、構造定義を無理に求めることはない。従って、構造定義類の選択には、悩まない方が得策である[17]。

4.3 TEI

独自に XML アプリケーションの開発をするのではなく、既存の XML アプリケーションの中から検討を行うのであれば、その候補として TEI は有力な選択肢である。

4.3.1 TEI とは

TEI 活動の詳細は、*Text Encoding Initiative*[7]、*TEI P5*[13] で解説されている。また、日本における TEI 活動の状況については、『SGML の活用』[14] や、本会議の土屋俊先生による発表で知ることができる[18]。

TEI の活動は、主に 2 つの目標に対して行われている。ひとつは、SGML/XML の不備に応える記述手法を検討すること、もうひとつは、各種テキストタイプ (e.g. ドラマ、詩歌、小説、etc.) 別の語彙集合 (スキーム) を検討することである[19]。XML アプリケーションを検討する様々な活動の中で、TEI を高く評価したい点としては、1) 人文科学資料を対象としていること、2) マークアップ言語一般の論議が行われていること、3) 多くの記述実験が行われていること、4) その結果、豊富な語彙が用意されていること、5) 語彙が多い割には、高い汎用性のある語彙が用意され、また見かけよりも単純でフラットな構造を持つインスタンスに仕上がること、6)TEI ファイルを、ハブファイルとして使用できること、7) テキストタイプ別のタグ集合を選択的に使用する (モジュール) ことができること、8) 文字レベルでの記述が検討されていること、9) 各種構造定義規格に中立であること、10) ソフトウェアの検討も行われていること、11) 国際化・地域化を意識していること、などを挙げることができる。TEI の利点として、人文科学資料の交換形式を挙げることもできるが、これは主たる利点ではないと考えている。タグ付き資料が利用される特定コミュニティを超える汎用性を持った交換形式であるよりも[20]、アノテーションをよりよく記述できるマークアップ言語のアプリケーションである点を、重視したい。

TEI には、多くの関連文書が用意されているが、中でも中心となるのは、1)*Guidelines for Electronic Text Encoding and Interchange(TEI P5)*[13], 2)*Text Encoding Initiative: Background and Context*[7] の 2 冊である。1) は、TEI で策定された語彙の構造と意味を規定・解説したもので、TEI 活動の集大成になっている。現在は、P4(第 4 版) が正規版であるが、2006 年中には P5(第 5 版) が正規版として公開される予定である。ガイドラインそのものの解説、P4 から P5 への変更点、ガイドラインにはない新しい活動成果など

[17]少なくとも、構造定義類の規格は、どれかが支配的になるという想定ではなく、今後も多くの規格が提案され、その中から目的にあったものを選択してゆく、というスタンスで検討するのが賢明であろう。

[18]近況を日本語で紹介するものは、全く無かったが、本会議のアブストラクトは日本語で公開されており、ここから最新の活動状況を知ることができる。`http://coe21.zinbun.kyoto-u.ac.jp/tei-day/tei-day2006.html.ja`

[19]他にも活動は幅広く行われているが [13]、この 2 つが重要であると考えている。

[20]実際には、この程度の可搬性がないことが目下の課題である。

については、別稿に譲りたい。2) は、ガイドラインとしてまとめられていることが、どのような議論・背景を元に策定されているかを詳細に解説した論集になっている。ガイドラインでは解説されていない状況で TEI タグを使用したい場合、それが適切であるかを判断するために、そのタグがどのような目的で策定されているのかを本書から知ることで、よりよく判断することができる。

4.3.2 TEI の使い方

TEI の使い方としては、1)XML アプリケーションとして TEI のみを使用し、2)TEI ファイルをハブファイルとして使用し、3) 必要最低限のタグ集合を選択する、ことが考えられる。TEI では、テキストタイプ別にタグ集合を定義し、それをモジュールとして選択することができる。これにより、資料のジャンルに必要なタグ集合に集中して検討を進めることができる。

ところが、実際に TEI を使い始めようとすると、実は、そう簡単でない場合が多い。特に、日本文化を形成する資料を扱う際は、そうである。

TEI の利点として、国際化・地域化が意識されていることを挙げたが、実は、これは TEI が抱える課題でもある。TEI で検討されているテキストタイプには、かなり偏りがあり、多くは欧米文化圏 (敢えていえば英語文化圏)[21]のものである。また、テキストタイプとして適応可能なものがある場合にも、タグの使用例[22]が欧米文化 (英語文化) のものが採られている。そのため、日本文化を形成する資料をマークアップする場合には、1)TEI タグ規定に従い、英語文化圏に倣うか、2) 追加・拡張を行うかを選択する必要がある。1) の選択は、恐らくない[23]。日本で TEI を使用する場合には、2) を選択せざるを得ない。これは、全くのオリジナルタグ集合の作成ではないが、ほとんどオリジナルタグ集合の作成に近い活動が求められることになる[24]。

この場合、3 つの活動が必要となる。ひとつは、TEI の中核タグ集合を使用しながら、日本文化の資料に必要となる追加タグ集合を検討することである。さらに、それを TEI へ提案することである[25]。人文科学資料の電子化・マークアップデータ化は、単に従来の研究手法をより迅速に、深く進める為の新形式の資料を作成することだけに意義があるのではない。アノテーションという人文科学研究の根本的行為が、電子化の対象となるという、全く新しい研究領域に携わることである。これには、記述対象物としての資料が文化的に多彩である程、手法の一般化に貢献する。文学・歴史・言語圏に閉じた資料を作成することに価値があるのではない。文化を越えた、科学的研究としての共同作業が必要となる[26]。ここに、TEI をベースにした、個別のオリジナルタグ集合を検討・作成する価値がある。

もうひとつは、TEI ガイドラインの国際化を進めることである。現行 TEI ガイドラインには、ソフトウェア開発で検討されてきたようないわゆる「国際化」に対応するためのガイドラインが含まれていない。文化を形成する資料を対象にしたガイドラインであるから、この論議の困難さは容易に想像がつく。これは、TEI の今後の重要課題である。この論議から実質的な成果を得るには、各文化圏からのマークアップ活動報告が重要になってくる。

---

[21] この差異を評価する知識がなく、このような曖昧な記述になっている。

[22] これは、タグの意味規定とも関連する。

[23] 必要とするアノテーションが十分に記述できず、不正確なものとなる。また、データを共有したい研究者は、日本文化の基準でマークアップされることを期待している。

[24] おそらく、TEI 活動が日本で盛んにならなかった理由の一番が、この点にあると考えている。「ほとんどオリジナル」でも「全部オリジナル」でも、結局、現行マークアップ言語の備による困難に直面するのであれば、はじめから「全部オリジナル」を選択されるかもしれない。

[25] 2005 年 7 月から、江戸時代の古地図を TEI によるマークアップを始めたが [11]、あまりの検討項目の多さに、嬉しくなってしまった。成果物としての発表は当分ないだろうが、アノテーションとマークアップ言語の関係を探る材料としては、大変貴重な資料になると感じている。まるで、フィールドワークで新しい言語の聞き取りに成功した感じである。また、ドイツ文化圏との関連性を指摘され、内容や鑑賞・解釈を超えた、マークアップを通しての比較検討という新たな研究領域があることも確認できた。

[26] 個別言語の記述と一般言語学の相補的な関係と例えることができるだろう。

もうひとつは、日本でマークアップ活動がより普及するために必要な、基礎資料を用意することである。残念ながら、TEIガイドラインを含めて、日本語によるTEIの解説や、マークアップの解説は、あまり無い[27]。TEIが実際に人文科学資料のマークアップ化の現場で使用されるためには、まず、検討材料となる基礎資料を充実させる必要があると考えている。そのため、現在、次期ガイドラインP5の翻訳を計画している[28]。実際にTEIを使ってマークアップを行うには、オリジナルタグ集合の開発が、日本の場合、恐らく必須であることから、十分に深い論議を個人活動として行うのに十分な基礎資料を用意してゆきたいと考えている[29]。

4.4 提案のまとめ:構造とスキーム策定への注釈

本稿の主張では、構造とスキーム策定について、微妙なスタンスを採っている。構造については、flatな構造を紹介する一方で、TEIスキームの採用を提案している。スキーム策定については、オリジナルスキームの作成には困難が伴うので、TEIの採用を促す一方で、TEIに従う場合でもオリジナルタグ集合は策定する必要に迫られることを紹介している。現行マークアップ言語の特性から観察される事実として、現行マークアップ言語は、アノテーション行為にとっては不十分なマークアップ言語である。この観察から得られる提案が、flatな構造などになっている。これとTEIスキームとの関連は、構造の高階性の程度問題になる。スキームの開発については、1)TEIを採用せずオリジナルスキームはflatに作成する、2)TEIを採用しオリジナルスキームはflatに作成する、ことが本提案から考えられる。TEIを採用する理由は、現行マークアップ言語の特性から得られるものではない。TEIの利点については、本稿の先述部分と、本会議で他の発表から得ることができる。

5 さいごに

本稿では、マークアップ活動を始めると感じる困難さには、現行マークアップ言語に含まれているsyntax上のくせがその一因としてあり、これはマークアップ言語の本質に関わる問題で、大変に難しい課題であることを紹介し、それへの対処法として、1) オリジナルのスキームを作る場合の策と、2) 既存のXMLアプリケーションを採用する場合にはTEIの選択が薦められることを述べた。

日本でこのような会議が開催されたことに、幸せを感じている。本会議を日本に招致して頂いた京都大学のC.Wittern先生には、心から感謝を申し上げたい。日本で停滞していた、TEIや人文科学資料のマークアップ活動一般がより豊かになり、先人の先生方が努力されたことが実になるよう、これを契機にマークアップ言語とアノテーション行為を連携して捉える活動を積極的に行ってゆきたいと考えている。

[27] TEI Liteを含む解説の日本語訳が、次のサイトで読むことができる。http://www2s.biglobe.ne.jp/~Taiju/markup.htm

[28] 本会議のポスター発表をご参照頂きたい。

[29] TEI Liteからの普及活動を始めてはどうかという提案をされたことがある。TEI Liteの邦訳は既にあるにも関わらず、利用は進んでいない。また、実際に人文科学資料のマークアップを行うには、TEI本体にも日本文化に対応したタグ集合が用意されていない現状では、TEI Liteでは一層、物足りなさを感じるだろう。TEI本体の利用においても、オリジナルスキームの作成という、構造の問題を含む困難な論議が必要になる。日本においては、TEIガイドライン本体の普及活動が必要であると考えている。また、簡易文書用には、DocBookがコンピュータ関連のコミュニティで既に普及している。TEIは、人文科学資料向けのタグ集合として捉えた方がよいのではないか。

参考文献

[1]S.Abiteboul, P.Buneman, and D.Suciu, 2000, *Data on the Web*, Margan Kaufman Publishers

[2]L.Burnard, K.O.O'Keeffe and J.Unsworth eds., (2006), "Electronic Textual Editing", Modern Language Association of America, forthcoming

[3]J.Cowan and R.Tobin, 2004, *XML Information Set(Second Edition)*, W3C

[4]J.Engelfriet and H.J.Hoogeboom, 1999, "Tree-Walking Pebble Automata", in J.Karhumäki et.al. eds *Jewels are forever, contributions to Theoretical Comuter Science in honor of Arto Salomaa*, Springer-Verlag

[5]D.H.Hansson, 2004, Rails, `www.rubyonrails.org`

[6]H.Hosoya and B.C.Pierce, 2000, "XDuce: A Typed XML Processing Language", Int'l Workshop on the Web and Database(WebDB2000)

[7]N.Ide and J.Véronis, 1996, *Text Encoding Initiative: Background and Context*, Kluwer Academic Publishers

[8]ISO, 1992, *Information technology – Hypermedia/Time-based Structuring Language(HyTime)*, ISO/IEC 10744, ISO

[9]F.Neven, 2002, "Automata theory for XML researchers, *ACM SIGMOD*, vol.131, No.3, ACM

[10]F.Neven and V.Vianu, 2004, "Finite State Machines for Strings Over Infinite Alphabets" *ACM Transactions on Computational Logic*, Vol.5, No.3, ACM

[11]K.OHYA, 2005, "A Progress Report on Encoding an Old Map in Japan", TEI 2005

[12]K.OHYA, (2006), "Active Usage of Obscure Content", draft

[13]C.M.Sperberg-McQueen and L.Burnard eds, (2006), *Guidelines for Electronic Text Encoding and Interchange(TEI P5)*, TEI Consortium, forthcoming

[14]根岸正光、石塚英弘編, 1994, 『SGML の活用』, オーム社

TEI: an Overview

Syd Bauman, Brown University

Introduction

It is easy to identify the Text Encoding Initiative with the Guidelines [01], whether you think of them as 1500 pages in two big blue books, or as 684 interconnected web pages. Either way, the Guidelines include advice on how to encode many things, some of which you may never have heard of, including elements for handling everything from linguistic terminology to poetry to historical events.

There are two potential problems with this. First, the TEI is more than just the Guidelines — it is also an organism, a research effort, and most importantly a community. Second, this wealth of information can make the TEI seem intimidating, forbidding, a challenge to master, perhaps even an impediment. I hope to convince you that while learning the Guidelines can be a challenge, it is not nearly as hard as it might seem, and it is well worth the effort.

The TEI is two things:

1. an XML text encoding language; and
2. an international consortium that exists to develop, maintain, support, promulgate, and use that encoding language.

It is not an international standard in the technical sense: it has not been codified in a fixed and permanent way by an official body. Rather, it is a community standard: it attempts to express a community consensus about how to encode textual information for humanities research, broadly speaking. Thus it functions as a sort of a lingua franca. It is not complete or finished: it is an ongoing research effort, which is described in more detail below.

The TEI Language

The outward expression of the Text Encoding Initiative, from the users point of view, is the TEI Guidelines. The Guidelines include a set of schemas, which can be thought of as formal rules for encoding documents, and a 1500-page document which explains how to apply these rules.

These Guidelines are not intended to be applied in their entirety to every document: they are not intended to make simple things difficult. Instead, they define a language, a descriptive lexicon which can be used in simple ways and also in complex ways, depending on the needs of the user.

In cases where it is important that the TEI Guidelines be applied strictly and in accordance with accepted practice, it is possible to constrain them closely. But in cases where it is more important to be able to express local nuances and specific details, the TEI also lends itself to more flexible encoding.

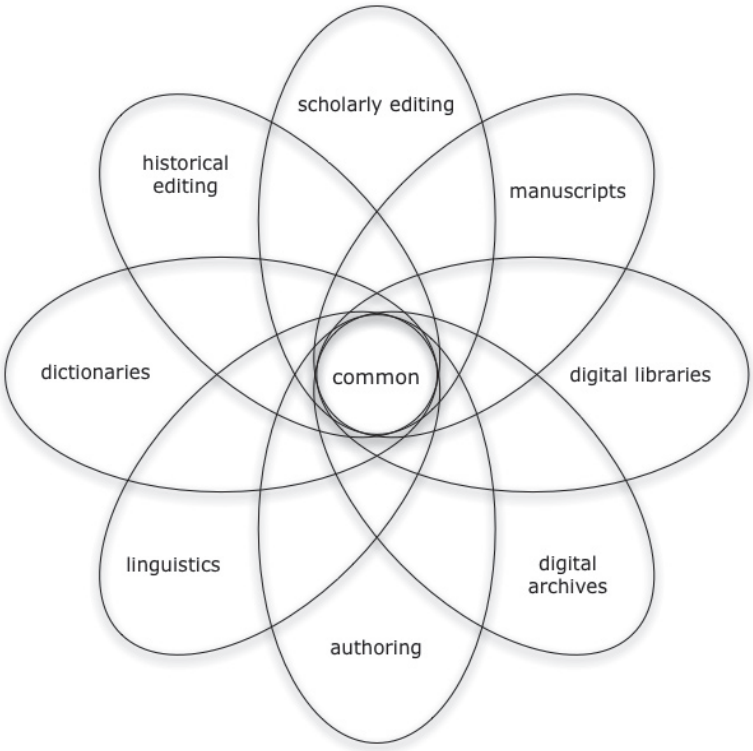
In this it is similar to a natural language: the TEI provides a very wide range of options, a variety of different ways of saying things; but it also provides a core of accepted rules which tend not to vary. Similarly, in formal contexts, where explicitness is essential (for instance, diplomatic, legal, or scientific language), formal constraints in natural languages tend to be strong, but in other contexts (for instance, poetry, humor, conversation, or private writing) the language rules are looser and more accommodating. For the TEI, in the same way, it is possible to apply the Guidelines in a highly constrained manner and with a great deal of explicitness. But it is also possible to apply the TEI more informally, for private purposes or for documents where the stakes are comparatively low.

People who need to talk to one another frequently tend to agree on how to express things: in other words, local usage serves as an additional guide. In natural languages this is often manifested in local dialects or in jargon. In the text encoding universe, this is manifested when disparate encoding groups agree on how to encode particular textual features.

Thus, although it sounds sentimental to say so, the TEI is not only an encoding language but the community of people and projects who use it. This is literally true in that there exist well used forums for communication among TEI users, including mailing lists and face-to-face meetings. Furthermore, the TEI Guidelines are themselves the product of this community: it arises from, among other things, the ongoing research, interpretation, and application that is performed by the TEI community, and in particular by the sub-communities that make it up. Because it is community-driven, the TEI language and its accepted usage emerges from the actual practice of TEI users, as much as from the specification itself.

The various TEI sub-communities not only participate in shaping the development of the Guidelines, but they also develop more specific guidelines, schemas, and documentation that express how to use the TEI for specific purposes: for instance epigraphy or manuscript editing. Any given sub-community has particular needs, some of which are peculiar to it, and some of which are shared by other sub-commmunities. This explains the relationship between these local efforts and the larger TEI universe.

The Universe of TEI Disciplines



When you use the TEI, you don't typically work with the entire Guidelines: you identify the subset of the Guidelines that will be useful for your project's needs.

Similarly you do not attempt to make your encoding useful to all people everywhere: you locate yourself within the group of people who are doing the same sorts of things you are: within your discipline or within your institution, or for the particular purpose you are addressing. It is with the members of this particular discipline or institution, or the practitioners of this particular activity with whom you share ideas and possibly even documentation, schemas, training materials, processing tools, etc.

This makes sense because you have a lot more in common with these people than with other TEI groups because there is more mutual relevance. You are using the same parts of the TEI, and even more importantly, you're using them for the same reasons in the same way: you have the same semantics for the same elements, the same kinds of critical insight or analytical motivations to express.

We can thus think of the TEI not as a single unified language, but as a set of closely related languages, and also as a set of disciplinary communities that share a core set of goals but also have specific needs and interests.

How is the TEI Used?

The TEI Guidelines is put to use in dozens of areas across the globe, many of which we know about; but there are probably equally many projects using the TEI that we do not know about, some for purposes we may have never imagined. Here are some examples of areas in which the TEI is often used.

Digital libraries and digital archives These are characterized by broad, shallow encoding of metadata to locate records which themselves carry little or no internal markup. The metadata and support structures typically include the capability to identify and thus retrieve and extract text chunks by variety of variables: e.g. by genre or topic keywords. Many encodings include markup of basic textual structures (e.g. chapters, headings) or basic physical structures (e.g. pages).

Thematic collections These are typically collections of a single author's work, or of works from a single period, genre, or locale. They generally contain deeper, thematic encoding which is used to analyze the internal structures or content of the text, e.g. comparison of vocabulary, or the analysis of the frequency of textual features.

Scholarly editions These typically include markup of standard editorial features such as textual variants, commentary, and textual notes.

Manuscripts The encoding strategy for manuscript materials emphasizes features that will assist researchers in studying the manuscript both as an artifact and as a textual carrier. TEI is used to encode both detailed descriptions of manuscripts, and also actual transcriptions of manuscript materials. Transcriptions often include markup of source issues such as revisions, orthography, and handwriting; as well as transcriptional issues such as illegibility or damage to the physical witness.

Dictionaries The TEI is used to mark up dictionaries in two ways: as historical documents in which the original details of representing the information is of primary interest, and as linguistic constructs in which the linguistic information being represented is of primary interest. It supports markup of the overall structure of a dictionary document, as well as the details of individual entries and their parts: headword, part of speech, definition, etymology, etc.

Language corpora The TEI includes support for markup of language corpora, including the encoding of disparate texts or portions thereof as a coherent whole; demographic information about speaker or author; additional metadata about the original delivery of a text; part of speech markup; syntactic markup; etc.

Historical documents, documentary editing This is a special genre of editing which is poised in between manuscript editing and scholarly editing, and also has some things in common with thematic collections, since it deals with primary source documents. Markup strategies for documentary editing include encoding of dates, names, and other data to assist a reader in making sense of the document; markup for handling problems of illegibility or damage to the physical witness; markup of metadata for retrieval.

Epigraphy This is the study of ancient inscriptions (on stone, buildings, etc.), and typically makes use of markup that focuses on a number of areas including transcriptional issues, e.g. illegibility & conjectural readings, and detailed markup of names & other features of linguistic interest.

Authoring TEI is also used for authoring articles, monographs, reports, grant proposals, letters, web sites, papers, and slides. This may involve markup of structurally relevant components, and also potentially of any other features the author finds useful: keywording, versions, alternative ideas or wordings, etc.

Structure of the TEI Guidelines

From the user's viewpoint, the TEI Guidelines consist of several distinct parts:

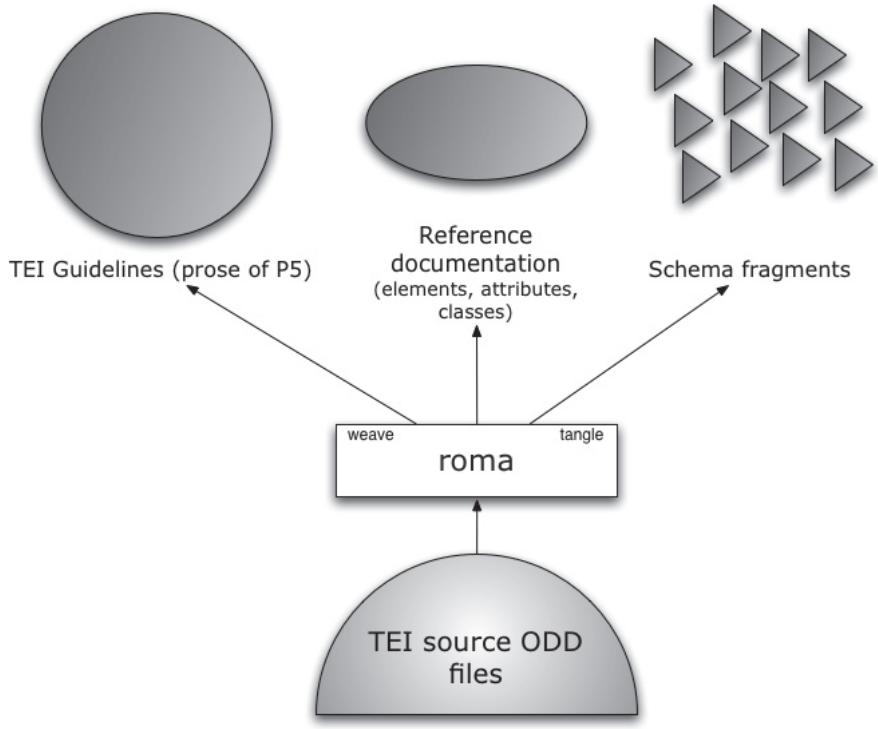
- the schemas that constrain the encoding of documents;
- the reference documentation for classes of elements, and the individual elements & their attributes that make up the schemas; and
- the text of the TEI Guidelines: the prose documentation which can be read on the TEI web site or in the big blue books.

These three things are actually all just products: forms of output. The TEI Guidelines are actually maintained as a single source which contains all of this information as one large (~3.2 MiB) document, which is then processed on demand to produce the various different parts as output.

That one document is stored in many separate system files for convenience, but it is still one XML document — which is, of course, written in TEI — from which all three outputs are derived. Hence the name “one document does it all” or “ODD”.

The utility we currently use to process the TEI Guidelines in their source (TEI “ODD”) format, extracting the various outputs is called *roma*. Note that the web tool *Roma*, or “Roma the web interface”, discussed further below, may be used as a convenient front-end to this command-line tool.

ODD: creating P5 itself



The ODD source of the TEI is divided into modules, or groups of elements, which constitute different functional components of the encoding system:

- the core elements which are needed by all documents
- the specific modules for different genres, such as verse or drama
- the specific modules needed for special types of texts, such as dictionaries or manuscripts
- the modules needed for particular specialized functions, such as linking, or encoding figures, or expressing uncertainty

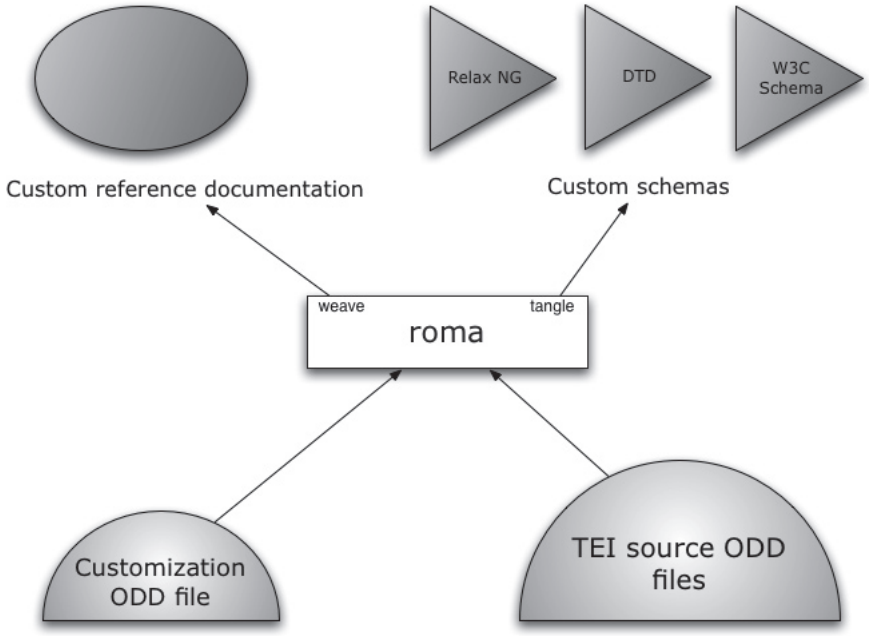
Customization

The TEI is not used directly in its raw form. In all cases a customized “view” of the TEI, or a “customization” is what is used. Such a customization includes both reference documentation and at least one schema tailored to the current particular requirements.

When users want to create a TEI customization, they create an ODD customization file that lists the modules they would like to use, the specific elements they would like to add or delete, the attributes they want change, etc. While this customization file is not particularly hard to write by hand, creating one is made even easier by a web-based front-end-editor called Roma. Many thanks to Sebastian Rahtz, who, with the help of Arno Mittelbach, created this wonderful tool.

This ODD customization file is then processed, along with the ODD source file for TEI P5, by the same utility used to create P5 itself: *roma*. The output of this process is customized reference documentation and customized schema(s). These outputs do not include every element in the TEI universe, but rather only the elements and classes from modules that were included in the customization file. In addition they contain any additional elements, examples, or descriptions that were added in the customization file.

ODD: Customization



This arrangement expresses what is fundamental about the TEI. There is no single, canonical view of the TEI; all views of the TEI are customizations. Some of these customizations, such as TEI Lite, are very frequently used, and others are less so (for instance, the EpiDoc scheme). Some are unique and are created and used by a single person. This system makes it possible for the TEI to accommodate a wide range of needs, from very simple to extraordinarily complex encoding, each for a large variety of multiple disciplines.

The TEI Consortium

TEI History and Challenges

The TEI was initially launched in 1987. It was sponsored by several leading humanities computing organizations:

- the Association for Computers and the Humanities,
- the Association for Literary and Linguistic Computing, and

- the Association for Computational Linguistics.

These groups worked together towards the common goal of producing an international encoding language for humanities scholarship.

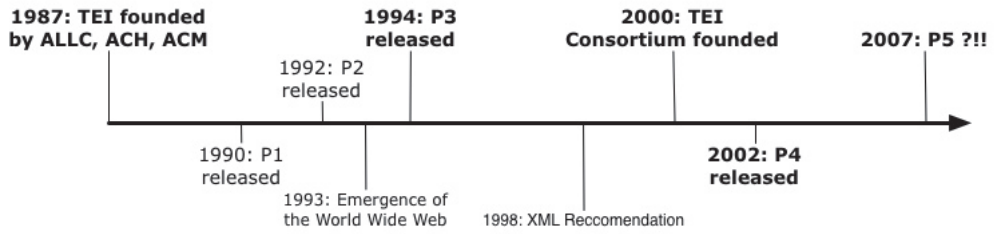
The main goal of this effort was to address an emerging Tower of Babel problem: the proliferation of a multitude of individual encoding languages. This problem brought with it the problem of mutual incomprehensibility. Thus, the goal was to create a single language for sharing encoded humanities data. Doing so would achieve several important benefits:

1. Reduce waste: both the waste of data itself through perishability due to formats that rapidly become obsolete, and the waste of effort in creating one-off encoding systems;
2. Increase longevity, so that important cultural information could be digitized with confidence that it will be readable and usable in the foreseeable future, thus preserving cultural heritage.
3. Increase our capability for interchange, research, collaboration.

However, even at the outset the effort faced several very significant challenges. First, it was deliberately aimed at an extremely broad community, that included the potential for use in many disciplines, in many languages, with diverse methodological assumptions: e.g., texts as linguistic objects vs texts as bibliographic objects vs. texts as literary objects. Furthermore, the intent was to cover a very ambitious set of possible documents: all humanities documents, plus related documents: social sciences, academic writing, etc.

Moreover, the system would need to permit any of a broad, heterogeneous set of assumptions about documents: e.g., the system needed to be able to capture the original appearance of a document, but also needed to be able to capture documents as data. Last, but not least, funding was slim. The TEI was operating on volunteer good will and institutional support.

The TEI development effort was funded by research grants from national funding agencies and private foundations: the US National Endowment for the Humanities, the European Union, the Canadian Social Science Research Council, and the Mellon Foundation.



The TEI worked within this model (intellectually sponsored by ACH, ALLC, & ACL, and funded by grants from various agencies & foundations) until 2000, releasing the first three versions of the TEI Guidelines. With the conclusion of that funding, though, the TEI needed a new organizational model: a way of supporting the development and maintenance of the Guidelines.

The new organizational model needed to make the TEI accountable to its users — to provide a structure that would allow it to be guided and controlled by those who use it. Thus, in 2000, the TEI Consortium was formed: a membership organization that would draw on the community of TEI users to support the TEI's work.

Organization of the Consortium

The organization of the TEI Consortium is rooted in the TEI community. The Consortium currently has 81 member institutions & individual subscribers [02]. It is hosted by four institutional hosts which contribute staff and funding: Nancy (coordinated by Loria, including ATILF and INIST), Oxford University, the University of Virginia (IATH and ETC), and Brown University (WWP, and the Library's Center for Digital Initiatives). The Consortium is guided by the TEI Board of Directors, which includes representatives from the hosts, and representatives elected by the membership.

The technical direction of the TEI is overseen by the TEI Council, which determines the future development of the TEI Guidelines, commissions work groups and reviews their work. Work groups are created and charged by the TEI Council to undertake specific areas of research: for instance, to write a new chapter on a new topic, or to examine the need for markup in a particular new area. Membership interests are also represented in Special Interest Groups, which may be organized on almost any topic, and which bring together members with similar interests to explore particular topics. Examples include TEI in libraries, TEI training, and the encoding of manuscripts. The two Editors are the focus for this development work: they make changes to the Guidelines, manage the overall coherence of the Guidelines, help keep the work groups focused, etc.

One can find details of the activity of the TEI on the TEI web site. Each work group and each SIG has a page, and the Council's activities are also documented there.

Annual Meetings

The most significant events of the year for the TEI are the annual Members' Meeting and the meetings of the TEI Council.

The Members' Meeting is officially where TEI business, in particular elections for members of the Board and Council, take place. But it is also an excellent forum where members, subscribers, and other interested parties meet to learn about each other's work, to meet with the TEI Board and Council members, and to learn about TEI work in some particular part of the world.

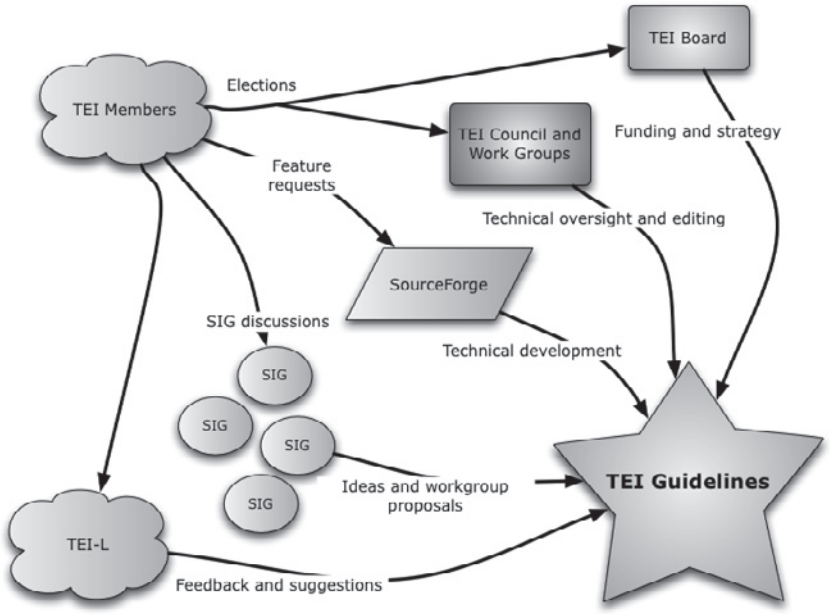
The 2005 Members' Meeting was in Bulgaria; the 2006 Members' Meeting will be held in Canada.

The TEI Council also meets annually, and this year is the first of what we hope may become a tradition, of associating a ‘TEI Day’ with that meeting. We hope this will serve as an opportunity for those using the TEI and those interested in doing so to meet, to exchange ideas, and to help each other.

We have already seen in this workshop the breadth of research that is being undertaken in this region: e.g., markup applied to the linguistics of languages with scarce extant textual materials and markup applied to corpora of spoken dialogs.

The most essential parts of the TEI community are the TEI projects and users. Their discussions and mutual assistance are conducted on TEI-L, the TEI discussion list. The list archives go back to January, 1990 and are a fascinating record of the TEI's history.

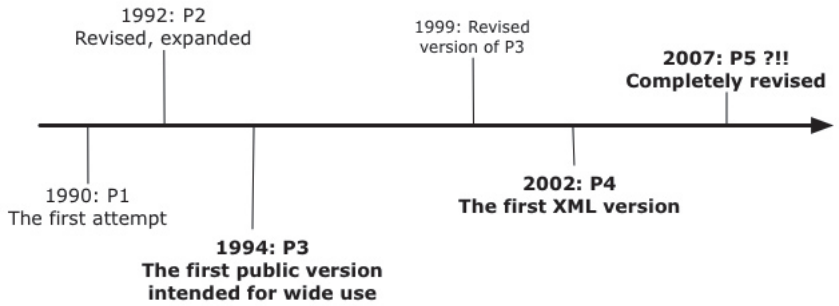
Ideas: from membership to Guidelines



It is important to mention, as long as we are talking about the TEI's organization, that all of the TEI's funding now comes from membership fees: both institutional and individual. Hence it is crucial for the TEI's ongoing survival and vitality that those who use the TEI, join the TEI.

Development of the TEI Guidelines

The TEI Guidelines did not spring forth like Athena fully armed from the brow of Michael Sperberg-McQueen and Lou Burnard; on the contrary, they have undergone a slow development process which is still not completed.



The first version of the TEI Guidelines was P1, issued in 1990. The second version, P2, was significantly amplified and contained many revisions and improvements; it was issued in 1993.

The third version, P3, was the first formal publication. It was much larger, more complete, and more usable; it is the first version that finally saw a lot of use. It was published in 1994, with a final revision in 1999.

P4, published in 2002, is the same as P3, but shifted from using SGML to using XML (or SGML). This was an important move, for a variety of reasons. First, it opened up the growing world of XML tools and put the TEI into the context of growing mainstream interest in text encoding. With the rise of XML and its increasing importance for the web, suddenly the TEI could be used for things like web publishing. This shift coincided with the huge growth in digital libraries and high-volume text encoding, which in turn made it reasonable to envision a less marginalized existence for the TEI. In fact, the TEI has taken center stage as a crucial component of scholarly digital publishing and digital libraries.

Because XML became ubiquitous, TEI's move to XML had another subtle but important impact on the usability of TEI in humanities computing. For the first time when you spoke to your friends and colleagues about TEI and what could be done with it, there was a good chance they would understand what you were saying. TEI was no longer the purview of humanities computing experts only. Scholars, researchers, web designers, and technologists could easily grasp it, too.

P4 is stable, well-documented, well-understood, and widely used. But of course we are not content …

P5

So naturally P5 is on the way. It is currently available as an alpha-test release, and being used by several projects. It includes much new material and substantial revisions of existing material to improve it and make it more consistent. Importantly, the theoretical underpinnings of the TEI scheme, the class system, has been completely revised.

Furthermore, because P5 can make use of XML schema languages other than DTDs, it introduces stronger formal data typing for attributes (and occasional elements), and hence it allows much tighter control over attribute values.

P5 is not backward compatible with P4, but migration of P4 documents will be primarily an automatic process. Migrating P4 DTD extensions to P5 will be somewhat more difficult, and there is not likely to be any software that performs this conversion — it will be a manual process. However, once you are using a P5 schema, managing user extensions and local customizations will be much easier

An initial release is planned for early 2007, subsequent releases with updates will appear thereafter. You can follow the development process and look at current drafts and releases on Sourceforge, where the complete source of P5 and the tools used to process it are publicly available.

Future Challenges for the TEI

The challenges the TEI faces result from its fundamental mission: to provide an international text encoding standard for a very broad range of humanities documents. And to do so with less funding than one might hope for.

One major problem for development of the Guidelines is to balance the need for improvement with the need for stability. Since the very beginning supporting research has been paramount. The TEI has been essentially a research project for its entire lifetime. It is instructional to note that the 'P' in the names P1 through P5 stands for 'proposal': intended to be tentative, experimental; not a finished product.

P3 was the first widely used version of the TEI, and it served as a testing opportunity. P4 did not change any of the substance of P3: it simply transformed it into XML. Thus the current version of the TEI is essentially the same as what was issued 12 years ago.

P5 is an attempt to act on what we have learned: to fix errors and smooth out inconsistencies, to supply some missing capabilities, and to bring the entire system together in a unified, coherent manner. P5 will represent a big break with the past, but once it is done, additional small changes can be made without as much disruption, including adding new elements that are needed, making small adjustments to how elements are defined, and similar updates.

Once P5 has been published, the TEI can focus more closely on important support activities. One of these is creating standard customizations for particular purposes: for instance, for manuscripts or for scholarly editions. Another important activity is providing documentation for particular communities and in particular for novice users who are trying to get started with the TEI on their own. We can sponsor the development of training materials and tutorials for different audiences and in different languages. The TEI is already engaged in translating the Guidelines and other materials, and this effort will continue well into the future. Finally, we can work on creating and gathering useful tools and stylesheets, and in general making the TEI easier to use.

How to get Involved, How to Learn More

There are a number of ways to become more familiar with the TEI and to involve yourself in the TEI community. One easy way to start is to subscribe to TEI-L. The best for the TEI is to become a TEI member or subscriber. (Note that the price is flexible, depending on what part of the world you live in and how big your project is. We want you to join!)

It is also possible to help with the P5 revision process. You can make suggestions using the Sourceforge feature request tracker. Suggestions made by 01 September 2006 will be considered for the initial release of P5.

Many people become involved by joining one or more of the special interest groups. If there is an area of interest for which there is no current SIG, you can start one of your own.

Visit the TEI website and the TEI wiki for more information about the TEI.

Acknowledgements

The author would like to thank his colleague Julia Flanders for her invaluable assistance in the preparation of the talk upon which this paper is based, and for the lovely graphics.

Notes

01 Sperberg-McQueen, C. M. and Lou Burnard, editors; *Guidelines for Electronic Text Encoding and Interchange*. March 2002. http://www.tei-c.org/P4X/

02 As of May 2006.

Towards an internationalized and localized TEI

Sebastian Rahtz, Oxford University

Abstract

The Text Encoding Initiative Guidelines have been widely adopted by projects and institutions in many countries in Europe, the Americas, and Asia, and are used for encoding texts in dozens of languages. However, the Guidelines are written in English, the examples are largely drawn from English literature, and even the names of the elements are abbreviated English words. We need to make sure that the TEI and its Guidelines are *internationalized* and *localized* so that they are accessible in all parts of the world.

The paper describes how the TEI project can develop internationally, including

- A review of why localisation and internationalisation matter
- A discussion of how the TEI architecture can be leveraged to support internationalised versions
- The application of the W3C ITS guidelines to the TEI work
- Practical results from a pilot project, and future translation plans
- The tools needed to make use of an internationalised TEI
- The steps towards ontologies in the TEI

1. TEI, internationalisation, and localisation

The Text Encoding Initiative Guidelines [TEI] have been widely adopted by projects and institutions in many countries in Europe, North America, and Asia, and are used for encoding texts in dozens of languages. For example, the projects listed at http://www.tei-c.org/Applications/ have examples of work involving Chinese, Danish, Dutch, Finnish, French, German, Greek, Hungarian, Italian, Japanese, Latin, Norwegian, Serbian, Spanish, Welsh, and some African languages; but given that the Guidelines are c. 1400 pages of fairly dense technical English, it is possible that only the more dedicated scholars get involved.

It may be useful to distinguish between what we might call 'traditional' or documentary approaches to translation, which focus on translating the descriptive prose of the Guidelines *as a document*, and 'formal' approaches which focus instead on translating the individual components (examples, element and attribute names, technical descriptions) in a way that enables these components to be used within the formal structures of the TEI as a technical standard. While the first approach may be very useful, the results are more difficult to maintain over the long term and are also more difficult to produce, since they cannot be accomplished in discrete chunks. The latter approach is the one we propose here, since it is more easily maintainable (only the affected elements need to be updated when changes are

made to the Guidelines) and can be more easily undertaken in a distributed fashion by collaborative groups.

Some translation work has already been undertaken:

- There have already been six 'traditional' translations of the TEI Lite (http://www.tei-c.org/Lite/) documentation into other languages. These have not covered translation of the element names or technical reference documentation. They are in wide use, however, and have created a need for more extensive translations of the Guidelines themselves.
- The French *Groupe d'experts n° 8* within CN 357 (*Commission de normalisation «Modélisation, production et accès aux documents»*) of the CG 46 (*Commission générale «Information et documentation»*) at AFNOR has an interest in TEI translation. Amongst other goals, this group intends to translate the definitions of the TEI elements and attributes into French. So far, they have worked in a 'traditional way' on some chapters of the P4 and P5 versions. Dissemination of the resulting French version of these chapters is very limited.
- Some 'formal' work has also been undertaken on translating element and attribute names; Alejandro Bia (for his background work see eg [BIA]) and Arno Mittelbach have prepared translation sets for Catalan, Spanish, and German. This work is integrated into the Roma (http://www.tei-c.org.uk/Roma/) application, allowing users to create tailored schemas in one of the supported languages.

Translation of documentation is only part of the issue. We need to make sure that the TEI and its Guidelines are *internationalized* and *localized* so that they are accessible in all parts of the world. The W3C define these processes as follows:

Internationalization (I18N)

> Internationalization is the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for redesign. Internationalization takes place at the level of program design and document development.

Localization (L10N)

> Localization is the process of taking a product and making it linguistically and culturally appropriate to a given target locale (country/region and language) where it will be used.

[http://www.w3.org/TR/itsreq/#intro_definitions]

Localization primarily concerns examples in the TEI context. There are over 1100 formal examples (that is to say, syntatically complete and valid) scattered through the text of TEI Guidelines, and another 775 in the formal definitions of elements; nearly all are in English. This is usually acceptable for examples like this:

```
Lexicography has shown little sign of being affected by the work of
followers of J.R. Firth, probably best summarized in his slogan,
<cit>
  <quote>You shall know a word by the company it  keeps.</quote>
  <ref>(Firth, 1957)</ref>
</cit>
```

which is in the field of discourse of many scholars, but many others require considerably greater familiarity with Anglo-Saxon culture. Even Shakespeare:

```
   <sp>
 <speaker>First Servant</speaker>
    <ab>O, I am slain! My lord, you have one eye left</ab>
```

```
    <ab>To see some mischief on him. O!</ab>
   </sp>
<stage>Dies</stage>
<sp>
    <speaker>CORNWALL</speaker>
    <ab>Lest it see more, prevent it. Out, vile jelly!</ab>
    <ab>Where is thy lustre now?</ab>
   </sp>
   <sp>
<speaker>GLOUCESTER</speaker>
    <ab>All dark and comfortless. Where's my son Edmund?</ab>
    <ab>Edmund, enkindle all the sparks of nature,</ab>
    <ab>To quit this horrid act.</ab>
   </sp>
```

is not easy, while older English is even harder:

```
   <lg>
    <l>Sire Thopas was a doghty swayn;</l>
    <l>White was his face as payndemayn,</l>
    <l>His lippes rede as rose;</l>
    <l>His rode is lyk scarlet in grayn,</l>
    <l>And I yow telle in good certayn,</l>
    <l>He hadde a semely nose.</l>
   </lg>
```

It will be countered that the words of these examples do not matter much, since it all that is required is to appreciate the markup constructs being used (most people will recall that Shakespeare wrote plays, and this is all that matters). However, sometimes the point of the markup is not obvious, as in this example:

```
Next morning a boy in that dormitory confided to his bosom friend,
a <distinct type="psSlang">fag</distinct> of Macrea's, that there
was trouble in their midst which King
<distinct type="archaic">would fain</distinct> keep secret.
```

Here there is the English word ‘psSlang’ (expandable to ‘public school slang’) for the type attribute of `<distinct>` to consider, where the value of ‘fag’ gives little help.

When the general context itself is clear, and the English text perhaps easy to translate, the names of the elements may stand in the way of easy comprehension. Thus:

```
   <persName key="EGBR1">
    <roleName type="office">Governor</roleName>
    <forename sort="2">Edmund</forename>
    <forename full="init" sort="3">G.</forename>
    <addName type="nick">Jerry</addName>
    <addName type="epithet">Moonbeam</addName>
    <surname sort="1">Brown</surname>
    <genName full="abb">Jr</genName>.
   </persName>
```

Can only really be take advantage of by someone who

1. appreciates the cultural context of ‘forename’ and ‘surname’

2. can mentally expan ‘nick’ to ‘nickname’ (and knows what a nickname is)
3. can appreciate whether a ‘Governor Edmund G. Jerry Moonbeam Brown Jr.’ is a politician, a kind of food, or a new dance

The user of the Guidelines may accordingly prefer to:

1. read ‘contiene un único documento TEI, compuesto de una cabecera TEI (TEI header) y un cuerpo de texto (text), aislado o como parte de un elemento corpusTei (teiCorpus)’ instead of ‘contains a single TEI-conformant document, comprising a TEI header and a text, either in isolation or as part of a teiCorpus element.’ in the documentation
2. use element names of `<líneaDirección>`, `<ligneAdresse>`, `<linDireccio>` or `<AdressZeile>` instead of `<addrLine>`
3. see examples from daily life, as in:

```
<Adresse>
 <AdressZeile>Herrn Jürgen Jemandem</AdressZeile>
 <AdressZeile>Computer+Software GmbH </AdressZeile>
 <AdressZeile>Albrecht-Thär-Straße 22</AdressZeile>
 <AdressZeile>48147 Münster</AdressZeile>
 <AdressZeile>GERMANY</AdressZeile>
</Adresse>
```

(thanks to http://www.columbia.edu/kermit/postal.html#germany for the example).

We will consider later how these translated element names can be reconciled with the English names.

It should be noted that element name translation by itself is quick and useful, but necessarily the most effective way to proceed. For example, many of the element names are in an abbreviated form of English (eg `<respStmt>`) which are not easy to translate sensibly. Furthermore, unless the reference descriptions are also translated, the element names by themselves do not give a clear idea of what the element is for. Using `<infoResp>` instead of `<respStmt>` is not as helpful as translating the description ‘supplies a statement of responsibility for someone responsible for the intellectual content of a text, edition, recording, or series, where the specialized elements for authors, editors, etc. do not suffice or do not apply.’

2. TEI architecture support for the I18N and L10N process

2.1. Unicode

The first priority in internationalizing the TEI is to ensure clean support for character sets throughout the system. With this in mind, the P5 revision of the TEI made substantial changes in its dealings with characters. As the W3C (http://www.w3.org/International/) recommend, in the TEI scheme:

- Unicode is the only supported character encoding schema. This means that entities for characters are deprecated, and the recommended daily use is for UTF-8 encoded text, as in

```
<persName xml:lang="el-grc">Φ λ . Θά λ λ ο ς</persName>
```

- There is a clean mechanism to use non-Unicode characters
- all appropriate text content models are set to allow a mixture of CDATA and <g> (where <g> is a reference to a non-Unicode character)
- all elements have an attribute xml:lang to record the language used
- there are no places where an attribute is used to hold pure text

A non-Unicode character can be defined using the <glyph> element in the TEI header. In the following example, we define a new character and assign it to a position in the Unicode Private Use Area (PUA); we also provide a standardized form as a fallback:

```
<charDesc>
 <glyph xml:id="z103">
  <glyphName>LATIN LETTER Z WITH TWO STROKES</glyphName>
  <mapping type="standardized">Z</mapping>
  <mapping type="PUA">U+E304</mapping>
 </glyph>
</charDesc>
```

This can now be referred to using the <gi> element, as in

```
<g ref="#z103"/>
```

At this point we will expect the processing application to work out what to do (either show the PUA character, if it can, or the standardized form). Other facilities in the <charDesc> element allow the user to provide an image file which has a picture of the character. It is also possible to override what appears in the text by using markup like this

```
<g ref="#z103">z</g>
```

where the content of the <g> element can be used immediately without any lookup.

Where a character is simply a relatively unimportant variant on a Unicode character, the user does not need to define a point in PUA, but can simply use <charDesc> to describe the variation.

2.2. TEI literate programming

The TEI is written in a high-level markup language for specifying XML schemas and their documentation. This language is an XML vocabulary known as ODD (*One Document Does it all*), and is one of the TEI modules. This provides a literate programming language for production and documentation of any XML schema, with three important characteristics:

1. The element and attribute sets making up the schema are formally specified using a special XML vocabulary
2. The specification language also includes support for macros (like DTD entities, or schema patterns), a hierarchical class system for attributes and elements, and the creation of pre-defined groups of elements known as modules.
3. Content models for elements and attributes are written using an embedded RELAXNG XML notation, but tools are available to generate schemas in any of RELAXNG, DTD language, or W3C schema.

4. Documentation describing the supported elements, attributes, value lists etc is managed along with their specification, together with use cases, examples, and other supporting material.

The expectation is that many people wish to use only a subset of the TEI, so the TEI's 22 modules (containing 500 elements) can be combined together and customized as desired using the ODD language, to produce a schema suitable for use by a project. Customization may including tightening the constraints on existing elements, removing unused elements, and even adding new elements or attributes (though this will make the text not portable).

The ODD language has allowance for translating element name, attribute names, and descriptions, and for preserving information to allow canonicalisation. The technical documentation elements (<gloss> and <desc>) for TEI elements and attributes etc can be specified multiple times, in different languages, distinguished by the standard xml:lang attribute. There is also a container (<equiv>) to specify the relationship of an element, attribute or value to standardised schemes.

Each definition of a new primary object (element or attribute) has associated description and examples. A complete example of a definition is as follows:

```
  <elementSpec module="header" ident="taxonomy">
   <gloss>taxonomy</gloss>
   <desc>defines a typology used to classify texts either
implicitly, by   means of a bibliographic citation, or explicitly
by a structured   taxonomy.</desc>
   <content>
    <rng:choice>
     <rng:oneOrMore>
      <rng:ref name="category"/>
     </rng:oneOrMore>
     <rng:group>
      <rng:group>
       <rng:ref name="model.biblLike"/>
      </rng:group>
      <rng:zeroOrMore>
       <rng:ref name="category"/>
      </rng:zeroOrMore>
     </rng:group>
    </rng:choice>
   </content>
   <exemplum>
    <egXML>
     <taxonomy xml:id="tax.b">
      <bibl>Brown Corpus</bibl>
      <category xml:id="tax.b.a">
       <catDesc>Press Reportage</catDesc>
       <category xml:id="tax.b.a1">
        <catDesc>Daily</catDesc>
       </category>
       <category xml:id="tax.b.a2">
        <catDesc>Sunday</catDesc>
```

```
      </category>
      <category xml:id="tax.b.a3">
       <catDesc>National</catDesc>
      </category>
      <category xml:id="tax.b.a4">
       <catDesc>Provincial</catDesc>
      </category>
      <category xml:id="tax.b.a5">
       <catDesc>Political</catDesc>
      </category>
      <category xml:id="tax.b.a6">
       <catDesc>Sports</catDesc>
      </category>
     </category>
     <category xml:id="tax.b.d">
      <catDesc>Religion</catDesc>
      <category xml:id="tax.b.d1">
       <catDesc>Books</catDesc>
      </category>
      <category xml:id="tax.b.d2">
       <catDesc>Periodicals and tracts</catDesc>
      </category>
     </category>
    </taxonomy>
   </egXML>
  </exemplum>
 </elementSpec>
```

The important things to note here are that the content model for the element is expressed in RELAXNG, which references other elements only by the names of classes to which they belong; and that the the worked example is well-formed XML embedded in its own namespace. This specification may be processed to produce a DTD, a RELAXNG schema, an XSD schema, or documentation in various forms.

The objects identified by the ident attribute in the TEI can be given an alternate name by use of the `<altIdent>` element; so the example above could be rewritten as

```
 <elementSpec module="header" ident="taxonomy">
  <altIdent xml:lang="fr">taxinomie</altIdent>
    ....
 </elementSpec>
```

Providing a French name for the element. How does this work in the schema, where other elements might refer to ‘taxonomy’? The normal schema, using RELAXNG compact syntax, has the definition

```
 taxonomy =
   ## (taxonomy) defines a typology used to classify texts either
   ## implicitly, by means of a bibliographic citation,
   ## or explicitly by a structured taxonomy.
   element taxonomy { taxonomy.content, taxonomy.attributes }
taxonomy.content = category+ | (model.biblLike, category*)
taxonomy.attributes = att.global.attributes, empty
```

in which the *element* <taxonomy> is defined by the containing pattern 'taxonomy'; it is the *pattern name* which other elements use, not the element name. If the schema were translated into Greek, it would look like this:

```
 taxonomy =
   element τ α ξ ι ν ο μ ι α { taxonomy.content,
taxonomy.attributes }
 ...
```

where the *pattern name* remains the same. This type of schema markup is generated by the TEI tools, picking up the information from <altIdent>. The descriptions work in the same way. We can expand the TEI source to add French translations alongside the English originals, and the appropriate text can be passed to the generated schemas or documentation:

```
  <elementSpec module="header" ident="taxonomy">
   <altIdent xml:lang="fr">taxinomie</altIdent>
   <gloss>taxonomy</gloss>
   <gloss xml:lang="fr">Taxinomie</gloss>
   <desc>defines a typology used to classify texts either
implicitly, by   means of a bibliographic citation, or explicitly
by a structured   taxonomy.</desc>
   <desc xml:lang="fr">L'élément Taxinomie <gi>taxonomy</gi>
définit     une typologie employée pour classer des textes soit
implicitement au     moyen d'une citation bibliographique, soit
explicitement au moyen     d'une taxinomie structurée.</desc>
  ....
  </elementSpec>
```

[We thank Pierre Yves Duchemin for these translations.]

What does a translated schema look like in practice? If we take a Spanish play, and translate the element names to Spanish (thank to Alejando Bia for this work), a text like this will be much more familiar-looking to encoders in Spanish-speaking countries:

```
  <cuerpo>
   <div1 tipo="part">
    <div2 tipo="act">
     <encabezado tipo="main">Jornada primera</encabezado>
     <div3 tipo="scene">
      <encabezado tipo="main">Cuadro único</encabezado>
      <acotacion formato="centered">
       <resaltado formato="bold">(Salen
</resaltado>REBOLLEDO,
     <resaltado formato="bold">la</resaltado>
CHISPA<resaltado formato="bold">
     soldados</resaltado>.<resaltado formato="bold">)</resaltado>
      </acotacion>
      <dialogo>
      </dialogo>
     </div3>
    </div2>
   </div1>
  </cuerpo>
```

This file will not work with normal TEI publishing tools, or be suitable for archiving, but it is straightforward to write a transformation (eg in XSL) which reads the TEI source with the element names and `<altIdent>` information, and puts the text back to canonical form.

2.3. TEI applications

TEI applications, as well as the texts themselves, need to have developed internationalised interfaces. For example, an application with turns TEI XML into HTML for web display, and provides a heading such as 'Contents' when it meets `<divGen type="toc"/>`, will have to provide appropriate translations. The TEI XSL family maintained by Sebastian Rahtz, for example (http://www.tei-c.org/Stylesheets/teic/), can operate in many languages:

ISO Language code	Text
en	Contents
de	Inhalt
ro	Cuprins
fr	Contenu
pt	Índice geral
es	Contenidos
slv	Vsebina
sv	Innehåll
sr	Sadržaj
pl	Spis treści
hi	Mula Shabda
nl	Inhoud
tr	İçerik
el	Περιεχόμενα

[Bulgarian, Chinese, Japanese, Russian, and Thai also available; they are omitted here to avoid printing problems.]

2.4. TEI schema-making tools

The ODD language files need to be processed to produce schemas in the chosen language. This is done by a set of XSLT scripts, which can either be run on a command-line, or as a web service called Roma (http://www.tei-c.org.uk/Roma/). This currently has support for varying the languages of its interface, but must also allow for supporting the following output schemes:

- canonical: English names, descriptions in English
- local descriptions: English names, descriptions in chosen language
- local names: names designed to make sense to a speaker of the chosen language, descriptions in English

- fully localized: both names and descriptions in chosen language

This work is in progress; while the underlying XSLT supports the generation of documentation in different languages, the web interface has still to be implemented.

2.5. The application of the W3C ITS guidelines to TEI work

An *Internationalisation Tag Set* working group (under the chairmanship of Yves Savourel, Enlaso) is writing a Recommendation (if it is accepted) for the World Wide Web Consortium about markup which encodes information for translators and localisors. The current state can be found at http://www.w3.org/International/its (this document is itself written using the TEI ODD language).
The ITS consists of a set of elements and attributes for annotating a text with information for further processing, covering **Internationalization:**

- Markup for bidirectional text
- Ruby annotation
- Language identification

and **Localization**

- Translatability of content
- The localization process in general
- Terminology markup

It is intended that the ITS annotation elements be added at several stages. The simplest is at the content authoring stage, by technical writers, developers of authoring systems, localizers or translators. In addition, specialist terminologists might annotation a text with terminological information, or localization engineers and translators may add information.
The primary ITS notion is that information about elements and attributes can be supplied

- in a document schema
- in an external rules file
- in a rule section in an instance file
- attached to instance elements

where the information consists of a set of *data categories*. On an instance element, for example, the following attributes may be attached

translate
: should this object be translated?

locInfo
: Is there some localisation hint?

locInfoType
: What type of hint is it?

term
: Does this object describe a technical term?

termRef
: Where is the term defined?

dir
: What is the text direction?

rubyText
: Is there some Ruby annotation?

A complete example of a TEI text marked up with a combination of ITS rules and ITS local markup looks like this:

```
<TEI xmlns:tei="http://www.tei-c.org/ns/1.0">
 <teiHeader>
 <its:rules>
  <its:translateRule translate="no" selector="//tei:body/tei:p"/>
 </its:rules>
 </teiHeader>
 <text>
 <body>
  <p>Hello <hi>world</hi>
  </p>
  <p
     its:translate="yes">translate me</p>
 </body>
 </text>
</TEI>
```

where the ITS rules say that <p> elements should not normally be translated, but the second <p> has an explicit override.

If we take a TEI ODD document, we can express the relationship between the structural elements and the documentation elements with the following ITS rules, which says that the default is to **not** translate anything, but gives a set of elements which **are** to be translated:

```
<its:rules xmlns:tei="http://www.tei-c.org/ns/1.0">
 <its:translateRule translate="no" selector="//tei:*"/>
 <its:translateRule translate="yes" selector="//tei:desc"/>
 <its:translateRule translate="yes" selector="//tei:gloss"/>
 <its:translateRule translate="yes" selector="//tei:valDesc"/>
 <its:translateRule translate="yes" selector="//tei:p[@rend='dataDesc']"/>
 <its:translateRule translate="yes" selector="//tei:remarks"/>
</its:rules>
```

Using this information, we can show graphically in Figure 1, Example of ITS implementation, using an ITS tool, which elements need a translated equivalent (those in green).

```
<elementSpec module="corpus" id="PERSON" usage="opt" ident="person">

 <equiv/>
 <gloss/>
 <desc>describes a single participant in a language interaction. </desc>
 <content>
  <rng:choice>
   <rng:oneOrMore>
    <rng:ref name="model.pLike"/>
   </rng:oneOrMore>
   <rng:zeroOrMore>
    <rng:ref name="model.personPart"/>
   </rng:zeroOrMore>
  </rng:choice>
 </content>
 <attList>
  <attDef ident="role" usage="opt">
   <equiv/>
   <desc>specifies the role of this participant in the group.</desc>
   <datatype>
    <rng:ref name="data.code"/>
   </datatype>
   <valDesc>a set of keywords to be defined</valDesc>
  </attDef>
  <attDef ident="sex" usage="opt">
   <equiv/>
   <desc>specifies the sex of the participant.</desc>
   <datatype>
Done
```

Figure 1. Example of ITS implementation

For the purposes of the formal translation procedure advocated by this paper, the ITS procedure provides a good framework.

3. Results so far

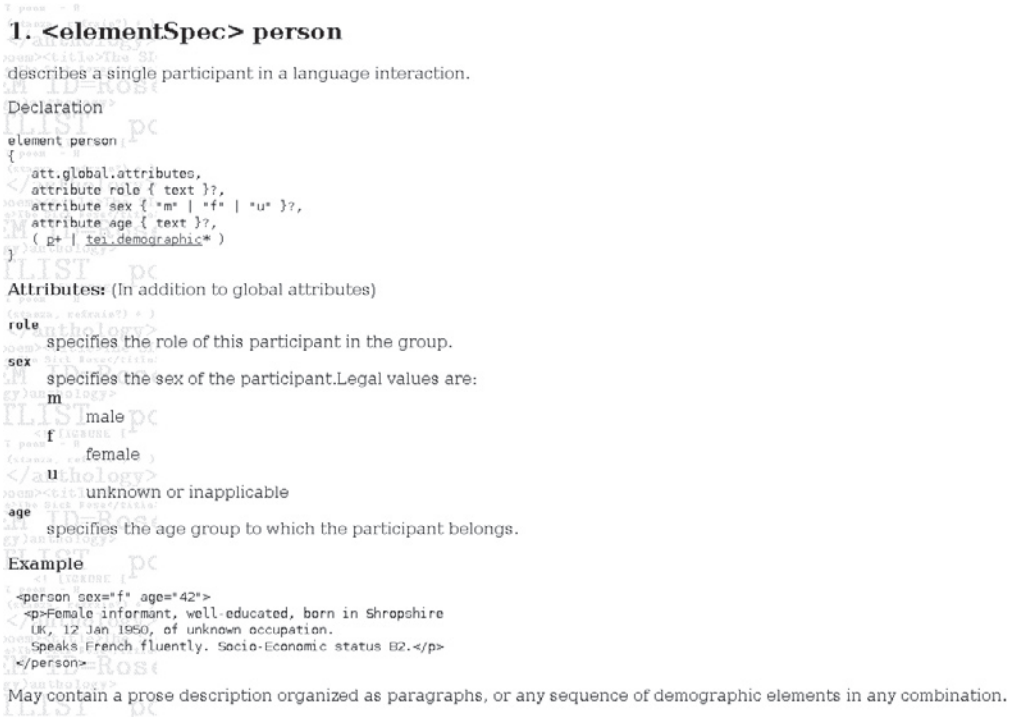
We present here some examples showing work completed so far:

```
<elementSpec module="corpus" xml:id="PERSON" usage="opt"
  ident="person">
 <equiv/>
 <gloss/>
 <desc>言語活動の関係者(1 件 1 名)</desc>
 <content>
  <rng:choice>
   <rng:oneOrMore>
    <rng:ref name="p"/>
   </rng:oneOrMore>
   <rng:zeroOrMore>
    <rng:ref name="tei.demographic"/>
   </rng:zeroOrMore>
  </rng:choice>
 </content>
 <attList>
  <attDef ident="role" usage="opt">
   <equiv/>
   <desc>当該関係者の言語活動における役割</desc>
```

```
     <datatype>
      <rng:ref name="datatype.Code"/>
     </datatype>
     <valDesc>定義済みキーワード</valDesc>
    </attDef>
    <attDef ident="sex" usage="opt">
     <equiv/>
     <desc>関係者の性別</desc>
     <datatype>
      <rng:ref name="datatype.Sex"/>
     </datatype>
     <valList type="closed">
      <valItem ident="m">
       <equiv/>
       <gloss>男性</gloss>
      </valItem>
      <valItem ident="f">
       <equiv/>
       <gloss>女性</gloss>
      </valItem>
      <valItem ident="u">
       <equiv/>
       <gloss>不明または不適切</gloss>
      </valItem>
     </valList>
    </attDef>
    <attDef ident="age" usage="opt">
     <equiv/>
     <desc>当該関係者の年齢層</desc>
     <datatype>
      <rng:ref name="datatype.Code"/>
     </datatype>
     <valDesc>推定年齢</valDesc>
    </attDef>
   </attList>
   <exemplum xml:lang="en">
    <egXML>
     <person sex="f" age="42">
      <p>Female informant, well-educated, born in Shropshire
               UK, 12 Jan 1950, of unknown occupation.
               Speaks French fluently. Socio-Economic status
B2.</p>
     </person>
    </egXML>
   </exemplum>
   <exemplum xml:lang="ja">
    <egXML>
     <person sex="f" age="42">
```

```
        <p>女性、教養あり、1950 年 1 月 12 日英国シュロプシア生まれ、          職業
不明、フランス語を流暢に話す、社会経済状態：中                    </p>
      </person>
     </egXML>
    </exemplum>
    <remarks>
     <p rend="dataDesc">
           段落単位の記述、または人口統計学のデータが混在して含まれる
         </p>
    </remarks>
    <listRef>
     <ptr target="#CCAHPA"/>
    </listRef>
   </elementSpec>
```

Figure 2. Example of translated ODD



1. <elementSpec> person

describes a single participant in a language interaction.

Declaration

```
element person
{
   att.global.attributes,
   attribute role { text }?,
   attribute sex { "m" | "f" | "u" }?,
   attribute age { text }?,
   ( p+ | tei.demographic* )
}
```

Attributes: (In addition to global attributes)

role
specifies the role of this participant in the group.

sex
specifies the sex of the participant.Legal values are:
m male
f female
u unknown or inapplicable

age
specifies the age group to which the participant belongs.

Example

```
<person sex="f" age="42">
 <p>Female informant, well-educated, born in Shropshire
 UK, 12 Jan 1950, of unknown occupation.
 Speaks French fluently. Socio-Economic status B2.</p>
</person>
```

May contain a prose description organized as paragraphs, or any sequence of demographic elements in any combination.

Figure 3. Example of reference documentation

1. <elementSpec> person

宣言

```
element person
{
   att.global.attributes,
   attribute role { text }?,
   attribute sex { "m" | "f" | "u" }?,
   attribute age { text }?,
   ( p+ | tei.demographic* )
}
```

属性: (グローバル属性の他)

role

sex

正当な値:

m

男性

f

女性

u

不明または不適切

age

例

```
<person sex="f" age="42">
 <p>Female informant, well educated, born in Shropshire
    UK, 12 Jan 1950, of unknown occupation.
    Speaks French fluently. Socio-Economic status B2.</p>
</person>
<person sex="f" age="42" lang="ja">
 <p>女性、教養あり、1950年1月12日英国シュロプシア生まれ、
        職業不明、フランス語を流暢に話す、社会経済状態：中
    </p>
</person>
```

段落単位の記述、または人口統計学のデータが混在して含まれる

Figure 4. Example of reference documentation in Japanese

1. <elementSpec> person

Декларация

```
element person
{
   att.global.attributes,
   attribute role { data.code }?,
   attribute sex { "m" | "f" | "u" }?,
   attribute age { data.name }?,
   ( p+ | model.personPart* )
}
```

Атрибути: (Освен глобалните атрибути)

role

sex

Разрешените стойности са:

m

мъж

f

жена

u

неизвестен или несъществуващ

age

Пример

```
<person sex="f" age="42">
 <p>Female informant, well educated, born in Shropshire
   UK, 12 Jan 1950, of unknown occupation.
   Speaks French fluently. Socio-Economic status B2.</p>
</person>
```

May contain a prose description organized as paragraphs, or any sequence of demographic elements in any combination.

Модул: corpus

Figure 5. Example of reference documentation in Bulgarian

Figure 6. Interface translation in Bulgarian

1. <elementSpec> person

Deklaration

```
element person
{
   att.global.attributes,
   attribute role { text }?,
   attribute sex { "m" | "f" | "u" }?,
   attribute age { text }?,
   ( p+ | tei.demographic* )
}
```

Attribute: (Neben global gültigen Attributen)

role

sex

Gültige Werte:

m

男性

f

女性

u

不明または不適切

age

Beispiel

```
<person sex="f" age="42">
 <p>Female informant, well-educated, born in Shropshire
          UK, 12 Jan 1950, of unknown occupation.
          Speaks French fluently. Socio-Economic status B2.</p>
</person>
<person sex="f" age="42" lang="ja">
 <p>女性、教養あり、1950年1月12日英国シュロプシア生まれ、
            職業不明、フランス語を流暢に話す、社会経済状態：中
          </p>
```

Figure 7. Reference documentation in Japanese, with German annotation

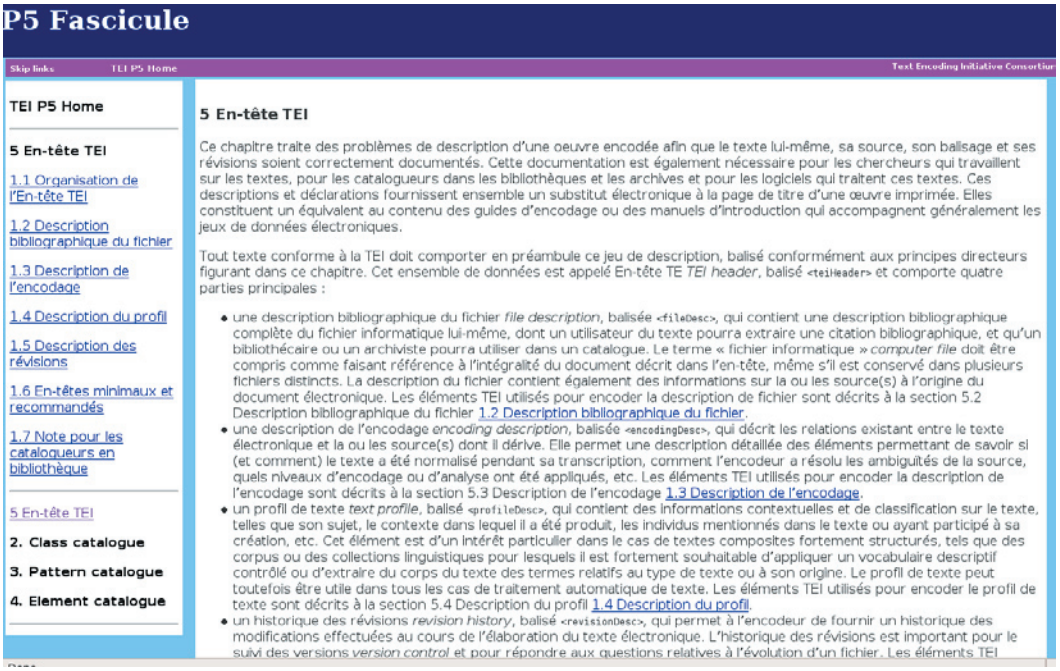


Figure 8. TEI Guidelines in French

4. Future directions

The TEI Consortium is working with TEI scholars to advance I18N and L10N in various languages (listed in). We hope to work on French, Spanish, German, Chinese and Japanese in 2006, and produce translated element and attribute names; translated `<desc>` and `<gloss>` texts, and a mechanism to allow users to easily take advantage of the work. The scale of work involved is not impossible to contemplate. The TEI contains

- 494 elements
- 489 attributes
- 1203 `<desc>` elements, 106666 characters
- 1177 `<gloss>` elements, 32385 characters

The worked needed for each language is to

- translate descriptive prose to other languages
- translate technical documentation components (note that this includes gloss for fixed attribute lists)
- translate examples
- localize examples
- add W3C ITS information
- translate processing workflow tool

The infrastructure challenges are not inconsiderable. We need, at least:

- An infrastructure to allow translators to submit material, and get prompt feedback
- Integrating the translations into the P5 source
- Ensuring that translations are flagged as decayed when the English original changes, and that translators are notified
- Managing multi-language examples

By the end of 2006, we expect to be well on the way to meeting these goals.

Appendix A Acknowledgements

The first steps in formalized internationalization of the TEI (as opposed to the translations of the Lite document) were made by Alejandro Bia, to whom many thanks are due. Translation examples in this paper come from Pierre Yves Duchemin (French), Marcus Bingenheimer (Chinese), Arno Mittelbach (German) and Alejandro Bia (Spanish). Veronika Lux and Julia Flanders co-wrote some of the explanations of TEI I18N.

Appendix B References

1. Manuel Sánchez, Alejandro Bia, Régis Déau, *Multilingual Markup of Digital Library Texts Using XML, TEI and XSLT*. Presented in XML Europe 2003
2. *The CIDOC Conceptual Reference Model* . Draft International Standard ISO/DIS 21127.
3. Christian Lieske and Felix Sasaki (eds). *Internationalization Tag Set (ITS) Version 1.0*. http://www.w3.org/International/its/itstagset/. World Wide Web Consortium, 2006.
4. Lou Burnard and Syd Bauman (eds). *Text Encoding Initiative Guidelines development version (P5)* . TEI Consortium, Charlottesville, Virginia, USA, Text Encoding Iniiative.

Appendix C TEI internationalisation partners

The following peoples and bodies have agreed to coordinate their respective languages:

Chinese	Marcus Bingenheimer	Chung-hwa Institute of Buddhist Studies, Taipei
Dutch	Bert Van Elsacker	-
French	Laurent Romary	Nancy
French	Veronika Lux	Nancy
German	Christian Wittern	Institute for Research in Humanities, Kyoto University
German	Werner Wegstein	Wuerzburg University
Hindi	Paul Richards	UGS (The PLM Company), http://www.ugs.com/
Hungarian	Király Péter	-
Italian	Fabio Ciotti	University of Roma
Japanese	OHYA Kazushi	Tsurumi University, Yokohama

Norwegian	Øyvind Eide	-
Polish	Radoslaw Moszczynski	Warsaw University
Portuguese	Leonor Barroca	Open University
Romanian	Dan Matei	CIMEC - Institutul de Memorie Culturala, România
Serbian	dr Cvetana Krstev	-
Slovenian	Tomaž Erjavec, Matija Ogrin	Dept. of Knowledge Technologies, Jozef Stefan Institute, Slovenia
Spanish	Manuel Sánchez	Miguel de Cervantes Digital Library
Swedish	Matt Zimmerman	NYU
Tibetan	Linda Patrik, Tensin Namdak	www.nitartha.org

XML markup of biographical and prosopographical data

M. J. Driscoll, Arnamagnæan Institute, University of Copenhagen

The charge

In the latter part of 2005 discussion began within the TEI Council as to how best to expand the capabilities of the TEI with regard to the mark-up of biographical and prosopographical data. It was formally decided in December to set up a work-group or 'activity' to look into this matter, which the present writer was invited to chair. According to its original charge (http://www.tei-c.org/Activities/PERS/persw01.xml), the TEI 'Personography Activity' was:

- to investigate in a systematic way other existing XML schemes used to handle data about people, extract a list of features they encode, and compare that feature set with the set currently supported by TEI P5.
- to review existing TEI customizations (e.g. Epidoc) which deal with people and work with their authors to identify features they have found it necessary to add.
- to review the existing TEI elements used to represent data about people, specifically those provided by the corpus and names and dates modules, review their coverage with regard to the feature sets identified in the first step and determine whether or not they should be presented as a distinct module.
- and, where there is a significant gap between the scope of other encoding schemes which cover historical data (e.g. HEML) and the current features provided by the TEI, to make recommendations on how the TEI scheme should relate to them.

In February 2006 Eva Wedervang-Jensen and the present writer produced a report comparing and evaluating a number of existing schemes for marking-up biographical and prosopographical data and comparing these schemes with the mechanisms currently available in TEI P5 (http://www.tei-c.org/Activities/PERS/persw02.xml). The schemes dealt with in the report were EAC (Encoded Archival Context), the CIDOC Conceptual Reference Model, MODS (Metadata Object Description Schema), METS (Metadata Encoding and Transmission Standard), EpiDoc (Epigraphic Documents), HEML (The Historical Mark-up and Linking Project), NOMEN, the GENTECH Data Modelling Project's XML implementation gdxml, GEDCOM XML, the HR XML Consortium's PersonName schema, and xNAL (OASIS TC's Name and Address Standard). Lists of features from these schemes were collected and fitted into ISAAR(CPF)2's standards schema (International Standard Archival Authority Record for Corporate Bodies, Persons and Families), which provided a ready-made matrix covering many relevant aspects of biographical data: names, dates and places of existence, nationality, occupation/sphere of activity, relationships etc.

At the end of April a meeting was held in Oxford, hosted by Oxford University's Classics Centre. The meeting was attended by Lou Burnard, Sebastian Rahtz and James Cummings from Oxford University Computing Services (http://www.oucs.ox.ac.uk/); Elaine Matthews of the Classics Department,

Oxford University, who runs the Lexicon of Greek Personal Names (http://www.lgpn.ox.ac.uk/) project; John Bradley and Gabriel Bodard of the Centre for Computing in the Humanities, King's College London, both of whom have been involved in a number of projects to do with prosopographical data, including EpiDoc (http://epidoc.sourceforge.net/), Prosopography of Anglo-Saxon England (http://www.pase.ac.uk/), Prosopography of the Byzantine World (http://www.pbw.kcl.ac.uk/) and Inscriptions of Aphrodisias (http://insaph.kcl.ac.uk); Fiona Oliver of the Ministry for Culture and Heritage of New Zealand, a production editor for *Te Ara – the Encyclopedia of New Zealand* (http://www.teara.govt.nz/), one aspect of which is a New Zealand Dictionary of National Biography. Also in attendance were Eva Wedervang-Jensen and the undersigned, both of the Arnamagnæan Institute, Copenhagen (http://www.hum.ku.kd/ami). At the meeting the report was discussed in the light of the actual experiences of the various participants and suggestions made for a general-purpose tagset for the mark-up of biographical and prosopographical data. What follows here is a presentation of the model developed at and immediately after that meeting.

Biography, prosopography and onomastics

A biography can be defined as an account of the series of events making up a person's life; this account will most probably make reference to other persons with whom the subject has had dealings, but the focus will normally be on a single individual. Prosopography, on the other hand, might be described as 'group biography', an investigation the characteristics of a group of actors in history by means of a collective study of their lives. One of the characteristics of a person will be his or her name(s), but it is also perfectly possible to identify 'actors in history' whose names are unknown to us (the 'tremulous hand of Worcester', to cite but one example). Onomastics is concerned with the origins and forms of proper names; for the onomastician individual persons are first and foremost bearers or indeed 'instances' of names; although their other characteristics — date and place of birth, social status, ethnic background and so on — will also be of interest, the focus of study is the name itself. At present the TEI provides some mechanisms for all of these, but there is for example no fully satisfactory way to record canonical information about the name itself as distinct from both its application to a person and the person to whom it is applied. One needs, in other words, to be able to mark up names encountered in source materials in such a way that they can point both to the person to whom that name refers (i.e. all references to that particular person no matter what form the name takes) and to the name as such (all instances of a particular name regardless of the identity of the person bearing it). The module described here is principally intended for use by those dealing with biographical and prosopographical data, but we recognise the importance of being able to accommodate the needs of omomasticians as well.

Users and uses

We envisage three basic types of users and uses to which this module might be put. The first, which might be called the 'DNB' model, is the person interested in creating or converting an existing set of biographical records, for example of the type found in a Dictionary of National Biography. The second, the 'DB' model, is the person hoping to create or convert a database-like collection of information about a group of people, possibly but not necessarily the people referenced in a marked-up collection of documents or a text-corpus; genealogists would also belong here. The third type, the 'CV' model, would be those interested in the creation or conversion of biographical or CV-like structured texts for use e.g. in Human Resources. In order to accommodate such a broad spectrum of users the

<person> element must be able to contain either structured components or plain prose; these structured components might also contain prose, but this prose should not recursively include structured components (except for very generic components such as dates or names). If one were interested, for example, in converting existing DNB-type records, and wanted to preserve the text as is, the <person> element could simply contain the text of an article, placed within <p> elements and using existing TEI elements such as <name> and <date>. For a more structured entry, however, one would extract the data and place it within the specific elements outlined below.

Basic principles

Information about people, we decided, whether of the DNB, DB or CV type, essentially comprises a series of statements or assertions relating to:

- personal characteristics or traits,
- states, and
- changes in state (i.e. events).

'Characteristics' or 'traits' are typically independent of an individual's volition or action and can be either physical, such as sex or hair and eye colour, or cultural, such as ethnicity, caste or faith. The distinction is not entirely straightforward, however: while sex is fairly obviously a physical trait, gender should rather be regarded as culturally determined, and the division of mankind into different 'races', proposed by early (white European) anthropologists on the basis of physical characteristics such as skin colour, hair type and skull measurements, is by many modern cultural anthropologists now considered to be more a social or mental construct than an objective biological fact. Furthermore, while some characteristics will obviously change over time, hair colour for example, none, in principle — not even sex — is immutable. 'States' include, for example, marital status, place of residence and position or occupation. Such states have a definite duration, and are typically a consequence of the individual's own action or that of others. By 'changes in state' is meant the events in a person's life such as birth, marriage or appointment to office; such events will normally be associated with a specific date or a fairly narrow date-range. Changes in states can also cause or be caused by changes in characteristics. Any statement or assertion on any of these aspects of a person's life will be based on some source, possibly multiple sources, possibly contradictory. Taking all this into account it follows that each such statement or assertion needs to be able to be documented, put into a time frame and be relatable to other statements or assertions of the same or any of the other types.

Existing TEI elements

There already exists in the TEI a mechanism for dealing with some of this data. This is the element <particDesc> (for 'participants description'), which contains one or more <listPerson> elements, each containing one or more <person> or <personGrp> elements, followed by an optional <particLinks> element, which can be used to indicate the nature of the relationship(s) between the individual persons. These elements form part of the module for linguistic corpora, and were originally intended to provide contextual information — such things as age, sex, geographical origins or socioeconomic status — on the participants in a linguistic interaction (see section 23.2.2 of the TEI Guidelines). Here is a typical example:

```
<person sex="2" age="42">
  <p>Female respondent, well-educated, born in Shropshire UK, 12 Jan 1950, of unknown
occupation. Speaks French fluently. Socio-Economic status B2.</p>
</person>
```

Because it was intended for this specific function, the existing <person> element is not appropriate for many of the kinds of users postulated above. Until recently there was, for example, a <birth> element but no corresponding element for <death>, since the participants in a linguistic interaction have obviously to be alive at the time their utterances are recorded. Similarly, there is an @age attribute on the <person> element for giving the age of the person at the time the interaction took place, which would not be appropriate for biographical or prosopographical data as age is obviously not a static thing. It was decided therefore to restructure and expand the existing mechanism and make it more generally available.

Proposed new elements

Three new classes of elements have been proposed, in keeping with the tripartite division outlined above: one for traits, one for states and one for events. Each of these classes contains a small number of specific elements for the most common types of biographical information, and a more general element for other, user-defined, types of information. All the elements in these three classes belong to the attribute class att.datable, which provides @from and @to, to indicate a specific date range, and @notBefore and @notAfter, to indicate the earliest and latest possible dates — the *terminus post quem* and *terminus ante quem* — for a characteristic, state or event. The value of all of these is a date in standard format as defined in ISO 8601:2000 5.2.1.1 to 3, extended format, i.e. yyyy-mm-dd. Most of these elements also belong to the class att.editLike, which provides @cert, @resp and @evidence, for indicating the degree of certainty, the source or agency responsible for the assertion, and the nature of the evidence supporting the assertion, and att.naming, which provides @key, enabling one to point to an externally-defined name for the referent and identify the scheme or standard used.

Characteristics

The class of elements for describing physical or socially-constructed characteristics or traits of a person contains the existing TEI elements <nationality> and <socecStatus> (for 'socioeconomic status'), in addition to which the following new elements have been proposed: <langKnowledge>, <faith> and <sex>. All, apart from <langKnowledge>, have a content model of macro.phraseSeq, by which is meant ordinary prose containing phrase-level elements.

```
<socecStatus key="#ab1" scheme="#rg">Status AB1 in the RG Classification
scheme</socecStatus>
```

The same information can usually also be expressed using empty elements:

```
<socecStatus key="#ab1" scheme="#rg"/>
```

<langKnowledge>, intended to replace the existing <firstLang> element, which was felt to be too restrictive, permits either paragraphs or a number of <langKnown> elements; both take a @tag (or @tags) attribute, which provides the standard code for the language as defined in RFC 3066 or its

successor, while <langKnown> also has a @level attribute to indicate the level of the person's competence in the language. It is thus possible either to say:

```
<langKnowledge tags="fu wo fr en"><p>Speaks fluent Fulani, Wolof and French. Some knowledge of English.</p></langKnowledge>
```

or

```
<langKnowledge>
  <langKnown level="fluent" tag="fu">Fulani</langKnown>
  <langKnown level="fluent" tag="wo">Wolof</langKnown>
  <langKnown level="fluent" tag="fr">French</langKnown>
  <langKnown level="basic" tag="en">English</langKnown>
</langKnowledge>
```

The <sex> element carries a @value attribute to give the ISO 5218:1977 values, i.e. 1 for male, 2 for female, 9 for non-applicable and 0 for unknown.

```
<sex value="2">female</sex>
```

In addition to these specific elements there is a generic element called <persTrait>, which has the content model label?, model.dateLike?, model.pLike*, (model.noteLike; | model.biblLike)*;, meaning it can contain an optional <label> element, which can be used to provide a human-readable specification for the category of trait or feature concerned, followed by a <date> or <dateRange> element, followed by the description of the feature itself supplied within one or more <p> elements, following which there may be added one or more notes or bibliographical references. The @type and @key attributes are available on the <persTrait> element to indicate a fuller definition of the combination of feature and value.

```
<persTrait type="ethnicity" key="#alb">
  <label>Ethnicity</label>
  <p>Ethnic Albanian.</p>
</persTrait>
```

Note that the generic element can be used in place of one of the more specific elements if greater flexibility is desired; <persTrait type="nationality"> is the same as <nationality>, but has more options in terms of content:

```
<persTrait type="nationality" key="#USA"/>
  <label>Nationality</label>
  <dateRange from="2002-01-15">From 15 January 2002</dateRange>
  <p>American citizen.</p>
</persTrait>
```

is the same as:

```
<nationality from="2002-01-15">Became an American citizen on 15 January 2002.</nationality>
```

or even:

```
<nationality from="2002-01-15" key="#USA"/>
```

States

The class of elements for dealing with states contains the existing TEI elements <persName>, <relation>, <occupation>, <residence>, <affiliation> and <education>, and a new <floruit> element for indicating the period during which a person is known to have been active, in cases where exact dates of birth and death are unknown:

```
<floruit notBefore="1066" notAfter="1100">fl. 1066-1100</floruit>
```

The <persName> element is repeatable and can, like all TEI elements, take the attribute @xml:lang to indicate the language of the content of the element, as well as a @type attribute to indicate the type of name, whether a nickname, maiden name, alternative form etc. This is useful in cases where, for example, a person is known by a Latin name and any number of vernacular forms, many or all of which may have claims to 'authenticity'. In order to ensure uniformity, the method generally employed in the library world has been to accept the form found in some authority file, for example that of the American Library of Congress, as the 'base' or 'neutral' form. Feelings can run high on this matter, however, and people are often reluctant to accept as 'neutral' an overtly foreign form of the name of their local saint or hero. Within the <person> element any number of variant forms of a name can be given, with no prioritisation, and hence less likelihood of offense. The Icelandic scholar and manuscript collector Árni Magnússon, to give his name in standard modern Icelandic spelling, is known in Danish as Arne Magnusson, the form which he himself, as a life-long resident of Denmark, generally used, and there is also a Latinised form, Arnas Magnæus, which he used in his scholarly writings. All three can be given, and in any order:

```
<person xml:id="ArnMag">
  <persName xml:lang="is">Árni Magnússon</persName>
  <persName xml:lang="da">Arne Magnusson</persName>
  <persName xml:lang="la">Arnas Magnæus</persName>
</person>
```

A large number of sub-elements is available within <persName> for the various parts of a name; here is an example using some of them (see section 20.1 in the Guidelines).

```
<persName>
  <forename sort="2">Sergei</forename>
  <forename sort="3" type="patronym">Mikhailovic</forename>
  <surname sort="1">Uspensky</surname>
</persName>
```

The generic element for states is <persState>, which has the same content model as <persTrait>. The example given here describes the first living held by the Icelandic clergyman and poet Jón Oddsson Hjaltalín:

```
<persState type="office" from="1777-04-07" to="1780-07-12">
  <p>Jón's first living — which he apparently accepted rather reluctantly — was at <name
type="place">Háls í Hamarsfirði</name>, <name type="place">Múlasýsla</name>, to
```

which he was presented on 7 April 1777. He was ordained the following month and spent three years at Háls, but was never happy there, due largely to the general penury in which he was forced to live — a recurrent theme throughout the early part of his life. In June of 1780 the bishop recommended that Jón should <q xml:lang="da">promoveres til andet bedre kald, end det hand hidindtil har havt</q>, and on 12 July it was agreed that he should exchange livings with <name

```
type="person" key="#ThorJon">sr. Þórður Jónsson</name> at <name
type="place">Kálfafell á Síðu</name>, <name
type="place">Skaftafellssýsla</name>.</p>
  <bibl>ÞÍ, Stms I.15, p. 733.</bibl>
  <bibl>ÞÍ, Stms I.17, p. 102.</bibl>
</persState>
```

Events

The class of elements for changes in state contains the existing TEI element <birth> and its recently-acquired sibling <death>. The generic element is <persEvent>; it has the same content model as <persState> and <persTrait> and can be used to describe any event in the life of an individual. In the example below the event is the wedding of Jane Burdon to the English writer, designer and socialist William Morris.

```
<persEvent type="marriage">
  <label>Marriage</label>
  <date value="1859-04-26">26 April 1859</date>
  <p><name type="person" key="#WM">William Morris</name> and Jane Burden were
married at <name type="place">St Michael's Church, Ship Street, Oxford</name> on <date
value="1859-04-26">26 April 1859</date>. The wedding was conducted by Morris's friend
<name type="person" key="#RWD">R. W. Dixon</name> with <name
type="person" key="#CF">Charles Faulkner</name> as the best man. The bride was given
away by her father, <name type="person" key="#RB">Robert Burden</name>. According
to the account that <name type="person" key="#EBJ">Burne-Jones</name> gave <name
type="person" key="#JWM">Mackail</name> <q>M. said to Dixon beforehand <q>Mind
you don't call her Mary</q> but he did</q>. The entry in the Register reads: <q>William Morris,
25, Bachelor Gentleman, 13 George Street, son of William Morris decd. Gentleman. Jane Burden,
minor, spinster, 65 Holywell Street, d. of Robert Burden, Groom.</q> The witnesses were Jane's
parents and Faulkner. None of Morris's family attended the ceremony. Morris presented Jane with a
plain gold ring bearing the London hallmark for 1858. She gave her husband a double-handled antique
silver cup.</p>
  <bibl>J. W. Mackail, <title>The Life of William Morris</title>, 1899.</bibl>
</persEvent>
```

In this example the keys on the various <persName> elements point to the <person> elements for the other people named. The <relation> element then be used to link them in a more meaningful way:

```
<relation name="spouse" mutual="#WM #JBM"/>
<relation name="parent" active="#RB" passive="#JBM"/>
<relation name="friend" mutual="#WM #RWD"/>
```

As mentioned above, all these elements, both the specific and the generic, are members of the att.datable attribute class, which means they can be limited in terms of time, as in the following example, where the person named David Jones has changed his name in 1966 to David Bowie:

```
<person xml:id="DB">
  <persName notAfter="1966">David Jones</persName>
```

```
  <persName notBefore="1966">David Bowie</persName>
</person>
```

The various attributes, `@value`, `@from`, `@to`, `@notBefore` and `@notAfter`, can be combined in various ways. Consider, for example, the following:

```
<date value="1857-03-15">15 March 1857.</date>
```

```
<date notBefore="1857-03-01" notAfter="1857-04-30">Sometime in March or April of 1857.</date>
```

```
<dateRange from="1857-03-01" to="1857-04-30">In March and April of 1857.</dateRange>
```

```
<dateRange from="1857-03-01" notAfter="1857-04-30">From the 1st of March to sometime in April of 1857.</dateRange>
```

All the generic elements are also members of the `att.editLike` class, which, as its name implies, was originally intended to provide attributes 'describing the nature of an encoded scholarly intervention or interpretation of any kind' and which makes available the attributes `@cert`, to indicate the degree of certainty, `@resp`, the agency responsible, and `@evidence`, the nature of the evidence used. In this way it is possible, in the case of multiple and conflicting sources, to provide more than one view of what happened, as in the following example:

```
<persEvent type="birth" resp="#XYZ" cert="high"><p>Born in <name type="place">Brixton</name> on 8 January 1947.</p></persEvent>
<persEvent type="birth" resp="#ABC" cert="low"><p>Born in <name type="place">Berkhamsted</name> on 9 January 1947.</p></persEvent>
```

Some examples

The Arnamagnæan Manuscript Collection

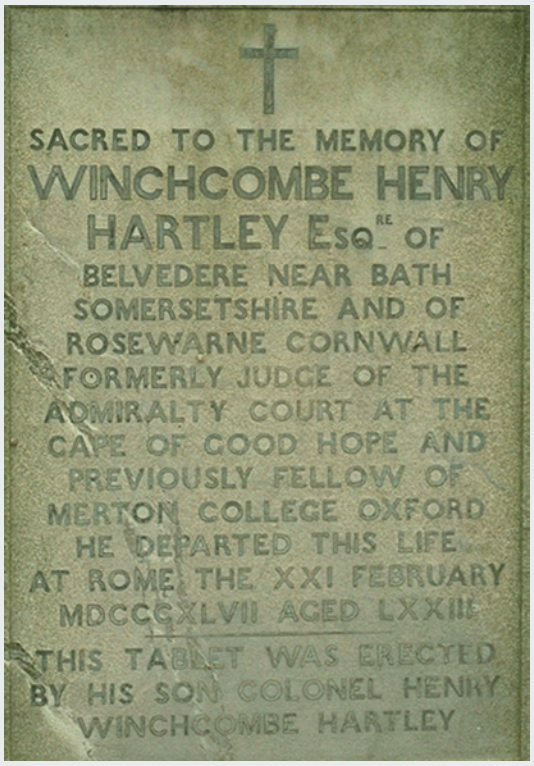
The Arnamagnæan Manuscript Collection is named after the Icelandic scholar and antiquarian Árni Magnússon (1663-1730), secretary of the Royal Danish Archives and Professor of Danish Antiquities at the University of Copenhagen. The collection, now divided between Denmark and Iceland, comprises some 3000 items, the majority of them Icelandic. The entire collection is in the process of being catalogued in TEI-conformant XML. In the catalogue there are named some 2000 persons (not counting biblical, historical and plainly fictional characters dealt with in the texts), the majority of them scribes or owners of manuscripts. We have tagged all these names using `<name>`, on which there is a `@key` attribute which points to the relevant `<person>` element in the header. Shown here is the `<person>` element for Jón Erlendsson, a seventeeth-century Icelandic clergyman who copied many manuscripts.

```
<person xml:id="JonErl01" role="scribe">
 <persName xml:lang="is">Jón Erlendsson</persName>
 <birth notBefore="1600-01-01" notAfter="1649-12-31">First half of the 17th century</birth>
 <death notBefore="1672-08-01" notAfter="1672-08-31">August, 1672</death>
 <sex value="1"/>
 <residence notBefore="1639-01-01" notAfter="1672-08-31">
```

```
  <placeName>
   <settlement type="farm">Villingaholt</settlement>
   <region type="county">Árnasýsla</region>
   <region type="compass">Southern</region>
   <country reg="IS">Iceland</country>
  </placeName>
 </residence>
 <occupation>Clergyman</occupation>
 <bibl><title>Íslenzkar æviskrár</title> III, pp. 105-106</bibl>
</person>
```

The Protestant Cemetary in Rome

Sebastian Rahtz has a large collection of data on the people buried in the Protestant cemetary in Rome. Here is one entry, and a photograph of the tombstone on which the information is based.




```
<person xml:id="WHH">
  <persName>
    <forename>Winchcombe</forename>
    <forename>Henry</forename>
    <surname>Hartley</surname>
  </persName>
  <sex value="1" resp="#SPQR"/>
  <birth notBefore="1773-02-22" notAfter="1774-
02-21" resp="#SPQR"/>
  <death value="1847-02-21">21 February 1847</death>
  <residence>
    <placeName>Belvedere</placeName>, near
<placeName>Bath</placeName>,
<placeName><region>Somersetshire</region></placeName>
  </residence>
  <residence>
    <placeName>Rosewarne</placeName>,
<placeName><region>Cornwall</region></placeName>
  </residence>
  <nationality resp="#SPQR">English</nationality>
  <occupation>Judge of the Admiralty Court at the Cape of Good
Hope</occupation>
  <occupation>Fellow of Merton College,
Oxford</occupation>
</person>
```

Topic Maps and TEI – using Topic Maps as a tool for presenting TEI documents

Conal Tuohy *conal.tuohy@vuw.ac.nz*
New Zealand Electronic Text Centre,
Victoria University of Wellington

Abstract

This paper describes a method used by the website of the New Zealand Electronic Text Centre (NZETC), in which Topic Maps are used as a tool for presenting TEI-encoded texts in HTML form.

Many electronic text archives transform their TEI texts into HTML for publishing their texts on the World Wide Web. Typically each chapter or page is transformed from TEI into a separate web page. Such a method produces websites that have the same structure as a physical book.

However, TEI is more expressive than HTML and can encode many other features of interest than just chapters, pages, and paragraphs. For example, TEI is also used to encode information about people and places and events, as well as literary criticism, and linguistic analysis. Indeed, TEI is designed to be extended to suit all kinds of scholarly needs.

These more complex aspects of text encoding are more difficult to transform into HTML. Because TEI is designed to be convenient for scholars to encode complex information, rather than for readers to understand it, it is necessary to transform the TEI into another form suitable for display. For instance, where a TEI corpus includes references to people, these references might be collated together to produce an index. For practical purposes, it is often necessary to extract information from TEI into a database, so that it can be queried conveniently and transformed into a web site.

The new "Topic Map" standard of the International Standards Organisation is identified as a suitable technology for solving this problem. A topic map is a kind of Web database with an extremely flexible structure. This paper describes a framework for using TEI in conjunction with Topic Maps to produce a large website which can be navigated easily in many directions.

What is TEI?

TEI (Text Encoding for Interchange) is an XML-based markup language (or a family of markup languages) for encoding texts. Although TEI is sometimes used to encode "born-digital" materials such as websites, more typically it is used to digitally encode the contents of printed books and manuscripts.

Multiple perspectives

A TEI document consists of a structured metadata header, followed by text, which may be broken up into paragraphs, and organised into parts or chapters.

However, going beyond its simplest form, TEI also offers a wide variety of specialized vocabularies for:

- linguistic analysis
- bibliographic metadata
- commentary
- dictionaries and thesauri
- biography and history
- … to mention a few.

Each of these vocabularies is designed to conform to some scholarly perspective or serve some scientific, literary, or other purpose. A TEI document which uses a combination of these vocabularies might therefore encode a multi-disciplinary, multi-dimensional description of a text.

TEI is not a presentation format

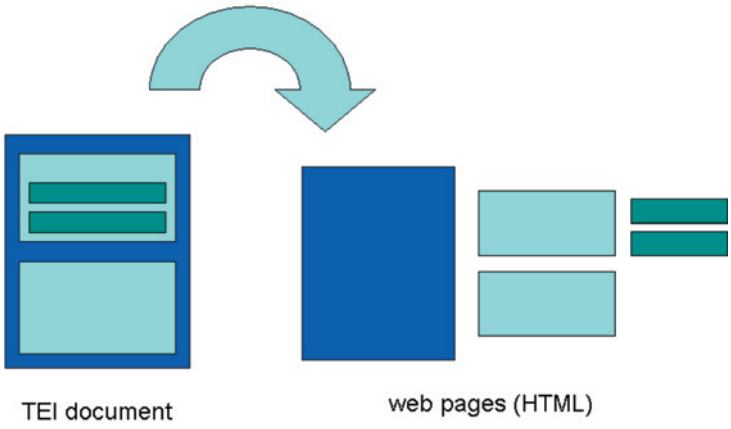
HTML is a simple presentation format in which text is described in terms of sections, headings, lists, and cross-references; similar to the concepts of word-processing. HTML is not generally concerned with the meaning of the text which it encodes. Because of this, generic web browsers can present HTML pages adequately no matter what their subject matter and content.

TEI, by contrast, is a format designed for text encoding in general, rather than just for presentation. TEI is much more complex than HTML, and can be extended even further to permit the description of texts according to whatever scheme or perspective a scholar wishes to use. The broadness and extensibility of the scope of TEI means that no single generic presentation mechanism can adequately present all possible TEI documents. Instead, any project which uses TEI must select, or develop, presentation mechanisms which are appropriate to the specific purposes of the project, and the specific types of encoding used in the project.

Data models and tools

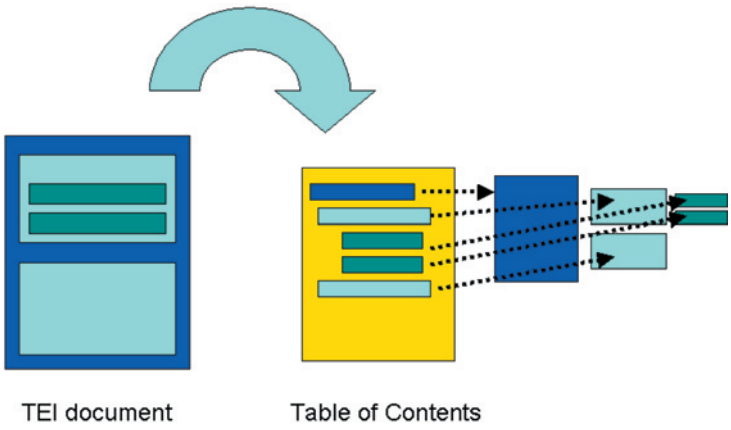
The currently standard practice for presenting TEI texts is to transform them into HTML, using Extensible Stylesheet Language Transformations (XSLT). XSLT is just one of a large number of XML-based technologies which we can use to model TEI texts as XML documents, with a tree-shaped structure.

Simple TEI documents have a structure which is comparable to that of HTML documents. These simple TEI documents can be transformed quite easy into HTML. Individual sections can be extracted and transformed into individual web pages.



Transforming TEI text into web pages

The tree structure of the TEI document can also be used to generate a web page containing a table of contents, with hyperlinks pointing to web pages which individually represent sections of the TEI. Similarly, tables of figures, lists of names, and other index pages can be generated by simple transformations of TEI into HTML.



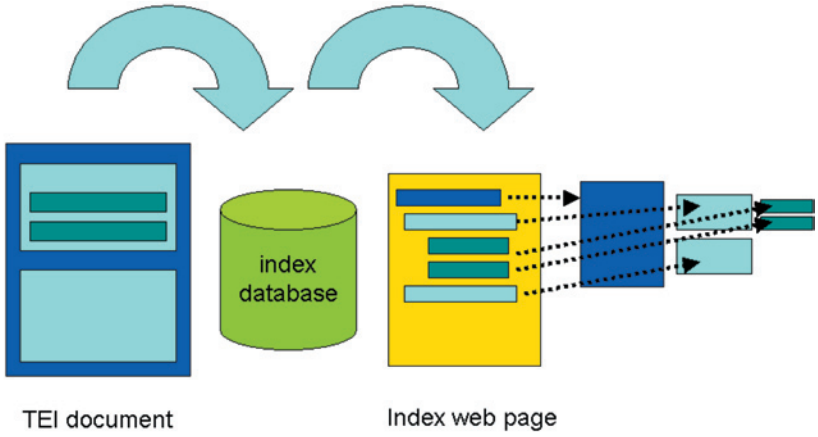
Transforming TEI structure into a table of contents

However, the complex data structures encoded in many TEI documents are often encoded using references which cut across the tree-structure of the XML. Generic XML tools are therefore not always ideal for processing these data structures, because they operate at a lower level of abstraction. Many of the more specialised concepts of TEI have no simple equivalents in HTML. Information encoded in this way can still be presented in HTML form, but it requires a more sophisticated transformation.

The markup in a TEI document is an expression of some theoretical model. Ideally, the software used to present the TEI-encoded text should also reflect that same model. However, because TEI embraces so many possible uses, it is a challenge to find software whose data model is really adequate to express all the different models which can be found in TEI documents.

Harvesting from TEI into a database

Some TEI markup can be hard to directly convert to an index, and often index pages will need to include material from a large number of TEI files
It is convenient to first extract (or "harvest") metadata from the TEI into a database, and then generate each HTML index from the database
For instance, where a TEI corpus includes references to people, these references might be collated together to produce an index. For practical purposes, it is often necessary to extract information from TEI into a database, so that it can be queried conveniently and transformed into a web site.



Harvesting TEI content into a database

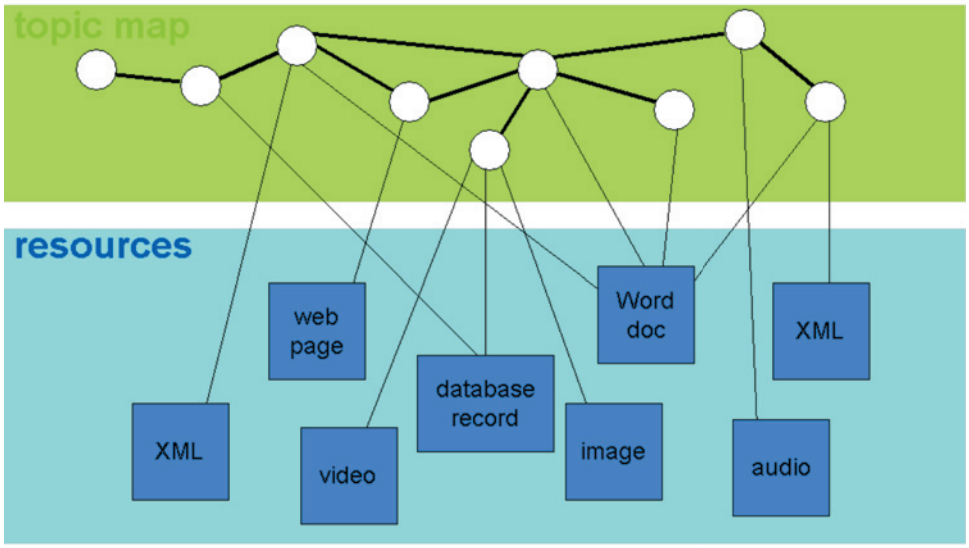
Topic Maps are an appropriate technology

At the NZETC we identified Topic Maps as a technology which could be used to model all the rich data structures which we had encoded in TEI.
A topic map is a kind of hyper-textual metadata. If a web site is a hyper-*text*, then a topic map is a hyper-*index*.
The Topic Maps standard is an evolution of an earlier technology called "Topic Navigation Maps", designed to represent and to merge the indexes of printed books. Topic Maps were adopted by the International Standards Organisation and became an international standard for knowledge representation and integration. In 2000, Topic Maps acquired an XML serialisation format: a markup language called "XML Topic Maps" or XTM. This XML formalism has greatly simplified the implementation of Topic Map solutions and facilitated the integration of Topic Maps with other XML technologies. Topic Maps are supported by a growing number of software implementations, both proprietary and open source.
Topic maps are very free-form, able to manage all kinds of metadata structures: catalogue records, indexes, tables of contents, controlled vocabularies, multiple hierarchies, glossaries, thesauri, and taxonomies, all can be linked together within a single topic map. The structural flexibility of topic maps is very important to us since this is necessary to model the features which are found in TEI documents.

Another valuable feature of Topic Maps is that they are based on standard web technology. They have an XML syntax and they use URIs for hyperlinking. This makes them a natural fit with XML-based TEI, and with the World Wide Web.

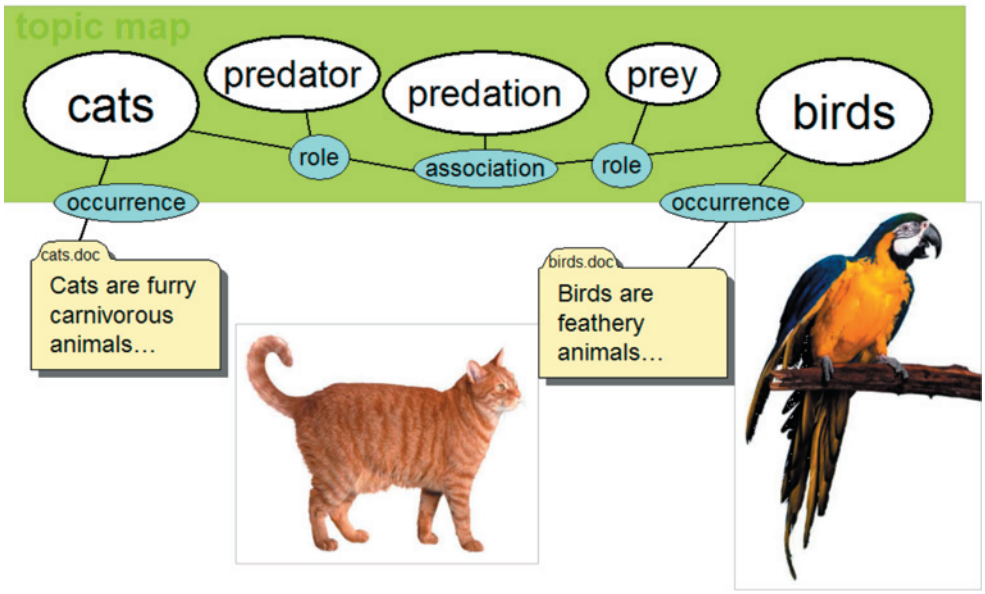
Topic Map key concepts

The following diagram shows how a topic map overlays a set of information resources, annotating them and defining relationships between them.
A topic map consists of a set of "topics" (the white circles in the diagram below). Each topic represents some subject of interest, which may represent a person, a document, a date, a page, a word, or whatever else is desired. Topics may be linked together by hyperlinks called "associations" (thick black lines in the diagram), and may also have associated information resources (i.e. documents of some sort) called "occurrences" (thin black lines in the diagram).



A topic map is a highly-structured repository of metadata which is distinct from the resources which it links to.

The diagram below shows a topic map containing topics called "cats" and "birds", which are intended to represent real cats and birds, respectively. The "cats" topic in the diagram has an *occurrence* called "cats.doc"; a resource which contains information about cats. Looking more closely at *associations*, we see that an association has a *type*, which is itself defined by a topic. Each member in the association plays a particular *role*, also defined by a topic. In the example, to model the fact that cats prey on birds, a *predation* association is defined in which cats play the *predator* role and birds play the role of *prey*.



Cats prey on birds

Building a topic map

We began our topic mapping by elaborating a conceptual model which would be adequate for the NZETC's digital library website.

First, we needed to decide which features of our TEI we were going to model in our Topic Map. Anything which we wanted to present on a page of its own had to be modelled as a topic.

One obvious set of elements were the TEI elements *group, text, front, body, back, div*, and *figure*. These elements, and the relationships between them, define the hierarchical structure of a book. We decided to create a topic in our topic map for every such element, which we called "TEI structural" topics. Where one such element contained another, we would also create a "containment" association between the corresponding topics in the topic map. This information would be used to generate tables of contents, and to provide next and previous links on each page.

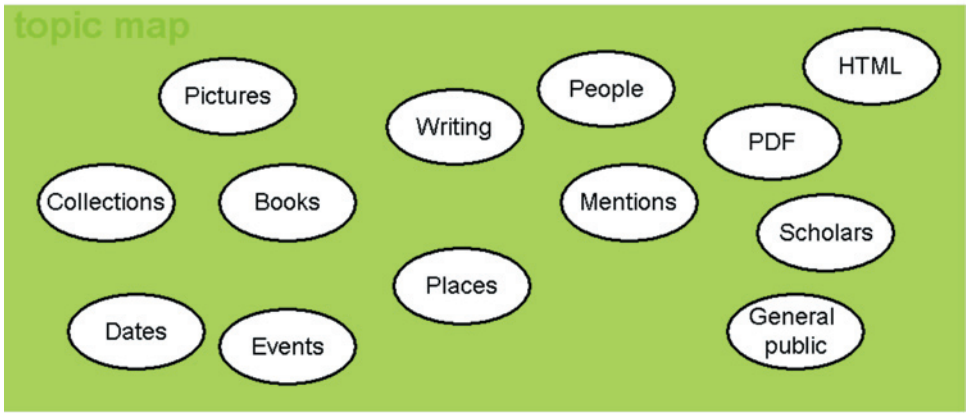
We also decided that we would create topics to represent people and places named in the texts using TEI *name* and *rs* markup. Initially this name markup was used for authors and publishers, but we gradually extended this to people mentioned in the body of the text as well.

Our XSLT harvester would create associations linking these named entities with the structural topics which mentioned them. This information would be used generate a web page for a person, containing links to all the places where the person is mentioned.

The other topics of interest were mainly derived from bibliographic data taken from the *teiHeader* metadata element. Such things as author, editor, funder, publisher, publication places and dates, subject classification, revisions, etc, could all be extracted from the TEI, represented as topics, and linked by associations to the topics representing the texts.

This analysis was the starting point for building our conceptual framework or ontology.

Ontology



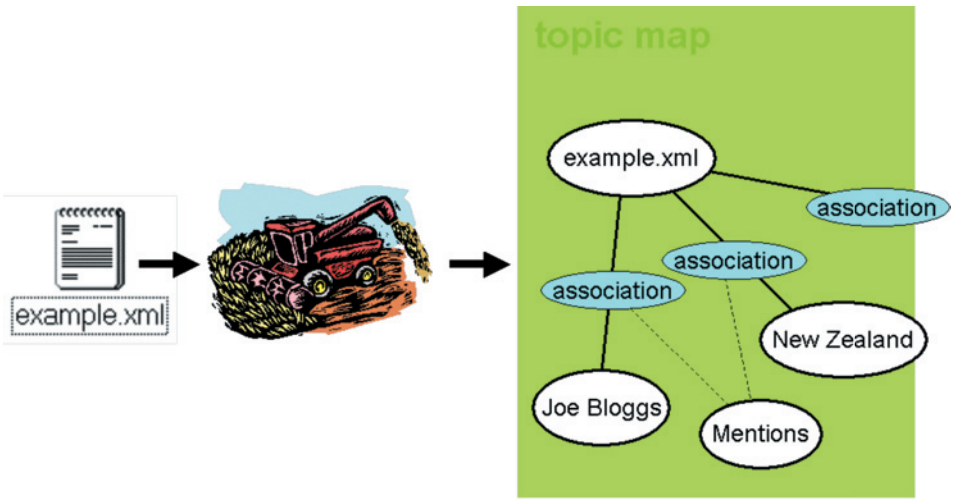
Ontology

To map your information, you first need to identify what kinds of subjects are of interest, what kinds of relationships these subjects may have, and what kinds of people would be interested (or what kinds of interest would they have).
For example, in our topic map the kinds of things we are interested in include books, pictures, people, and places.
The kinds of relationships we're interested in include: books mention people, people write books, books contain pictures, and people and places are depicted in pictures.
People might be simply interested in the content of texts, or they might have a scholarly interest in the physical details of the texts.
These concepts are what our site is all about. They form the conceptual foundation of a topic map, sometimes called an "ontology".
The term "ontology" as it is used here is borrowed from philosophy, where it refers to the branch of metaphysics which deals with the nature of being and existence. Computer scientists have dragged the word down from its philosophical heights, and it can now even be used simply to mean a computerised representation of a conceptual framework.
We could have just built an ontology of our own, from these concepts, but we felt we might discover conceptual inadequacies in our model later, so to play safe we linked it up to an existing ontology.
This ontology was produced by the International Committee for Documentation (CIDOC) within the International Council of Museums (ICOM), known as the CIDOC "Conceptual Reference Model".
The CIDOC ontology is quite large, and we only needed part of it, so we just selected the concepts we needed, used those, and ignored the rest. If and when we decide we need to expand our ontology, we'll just add some more of their concepts into our ontology.

Harvesting topic maps

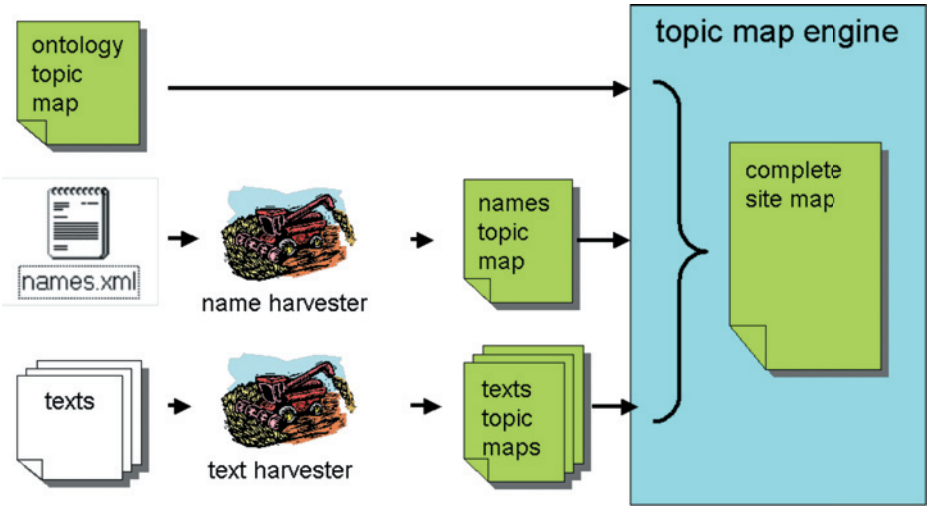
In our project we already had considerable experience with XSLT, which we'd been using to convert TEI documents into HTML, so we were able to put those skills to use writing XSLT to transform TEI

documents into Topic Maps. These transformations identify the topics of interest in a TEI document and the relationships between those topics.
To produce the final Topic Map, all these harvested topic maps will be merged together. This produces a map of our entire website, currently containing about 50 thousand topics in total.
At the NZETC, our source materials are all in XML, which makes it particularly easy for a harvester to find the things it's interested in – names of people and places, pictures, authors, etc. Our harvester takes only minutes to read through hundreds of megabytes of XML files, and harvest tens of thousands of topics and hundreds of thousands of associations from out of them.
By encoding our metadata in TEI we are able to edit it freely while retaining the benefits of schema validation. By automatically harvesting metadata from the TEI to maintain the topic map we can reliably keep the topic map accurate, detailed, and up-to-date.



Harvesting a topic map from a TEI XML document

Our TEI harvester is an XSL transformation which transforms the TEI document into an XML Topic Map document. The XSLT creates an XTM topic to represent each TEI document (and each major section of the document). The XSLT also creates XTM topic for each TEI name it finds, and creates a "mentions" association between the topic which represents the text and the topic which represents the thing named. In the example the topic map asserts that the TEI document mentions both Joe Bloggs and New Zealand.



Topic maps harvested and merged to form a single unified map

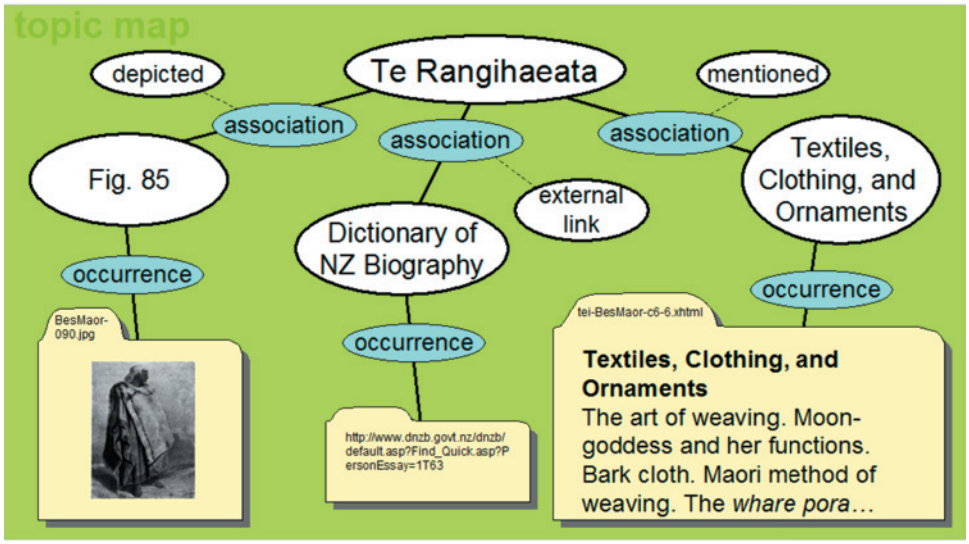
The final topic map is compiled by a topic map engine, by merging the topic maps from our three sources:

1. Our ontology topic map (containing the small set of our basic concepts).
2. The names map harvested from our name list (quite a large topic map)
3. Topic maps harvested from our XML texts (there are a couple of hundred of these, some big, some small)

As it reads each topic map, the topic map engine tries to find existing topics that match the topics in the map being imported. Where it finds an existing topic which represents the same subject as the new topic, the topic map engine automatically merges the two topics together. Topics can be made to merge automatically simply by sharing a unique identifier of some sort. This means that the data import process is declarative rather than procedural; it is simply enough to assert that two topics represent the same subject, and the Topic Map engine will merge them into a single topic, combining all the characteristics of the two original topics.

Creating a website from the topic map

Finally it only remains to display the topic map as a website.
To do this, we programmed our web server to generate a web page for each topic in the map. To do this, the web server asks the topic map engine for a topic, and creates a web page by copying information from the topic, as well as from topics which are associated with it, and from occurrences of those topics.



Fragment of NZETC website topic map

The diagram above represents part of our topic map. The central topic, Te Rangihaeata, was a chief of a Maori tribe, the Ngati Toa. In the topic map, the topic which represents him is associated with three other topics, each of which has an occurrence.

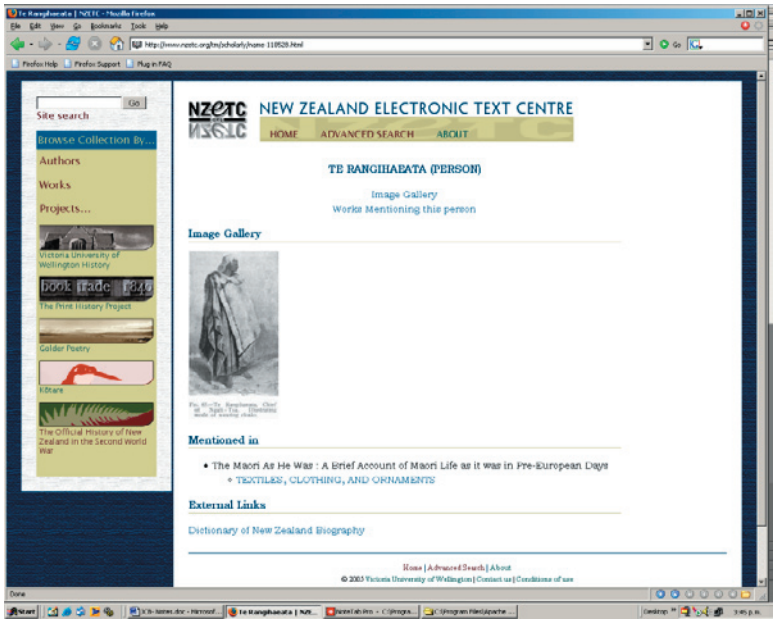
The association on the left represents a depiction of Te Rangihaeata. The picture which depicts him is "Figure 85" from a book by Elsdon Best called *The Maori as he was*.

On the right, "Textiles, Clothing and Ornaments" is a chapter from the same text, which mentions him. Both of these associations were of course harvested from TEI name markup in the TEI file in which *The Maori as he was* is encoded.

In the centre of the diagram, Te Rangihaeata is associated with a web page on the website of the Dictionary of New Zealand Biography. This last piece of information was harvested from our name list.

Note that the central topic "Te Rangihaeata" was harvested twice – once from the Elsdon Best book, and once from the names list. After harvesting, these two topics merged together automatically, leaving us with just one topic with 3 associations.

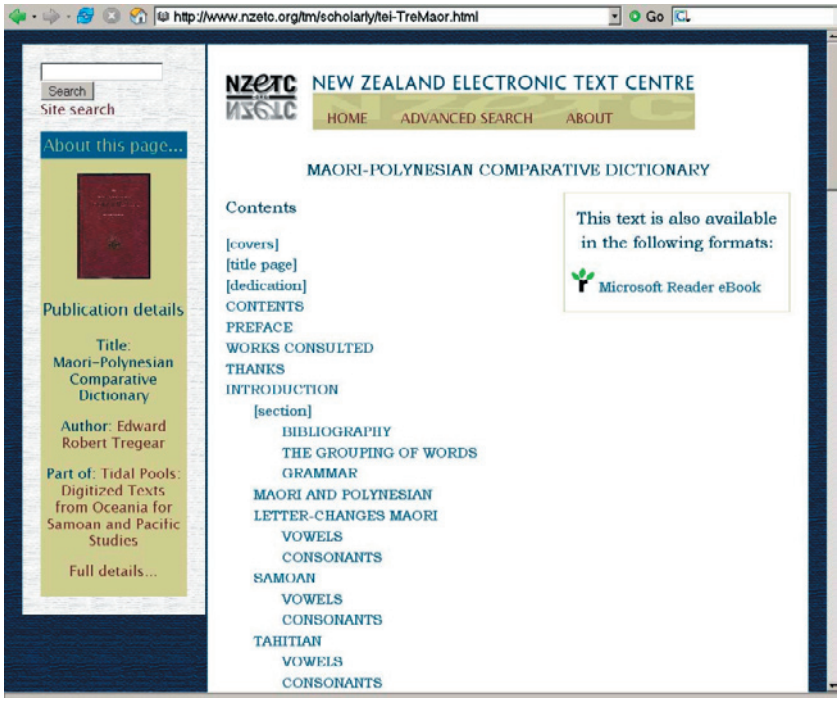
The figure below shows how the "Te Rangihaeata" topic is displayed as a web page. The page shows his name, the depiction, and the mention, as well as the external link.



Te Rangihaeata web page

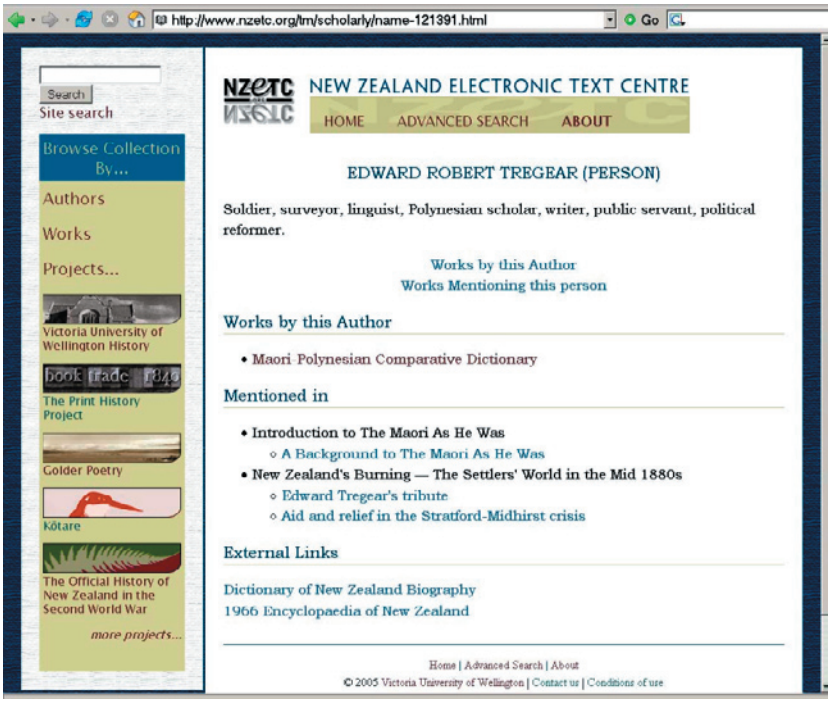
Topic Map supports exploration

The topic map underlying the website now allows for an exploratory style of navigation.
The web page below shows a dictionary written by a scholar called “Edward Tregear”. Notice that his name appears (as a hyperlink) in a sidebar on the left. This sidebar is visible not only here on the contents page, but throughout the book.
This hyperlink is an expression of an association in the topic map between the topic representing Edward Tregear, and the topic representing the writing of the dictionary. This association was harvested from a TEI *author* element in the *teiHeader* element of the text



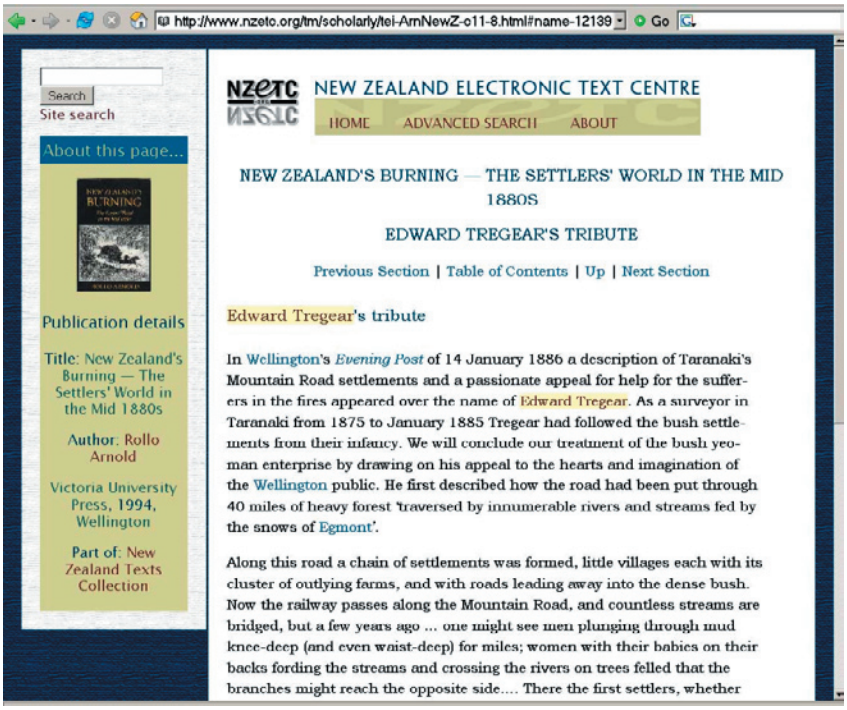
Edward Tregear's *Maori-Polynesian Comparative Dictionary*

Clicking on his Edward Tregear's name takes you to a page about him:



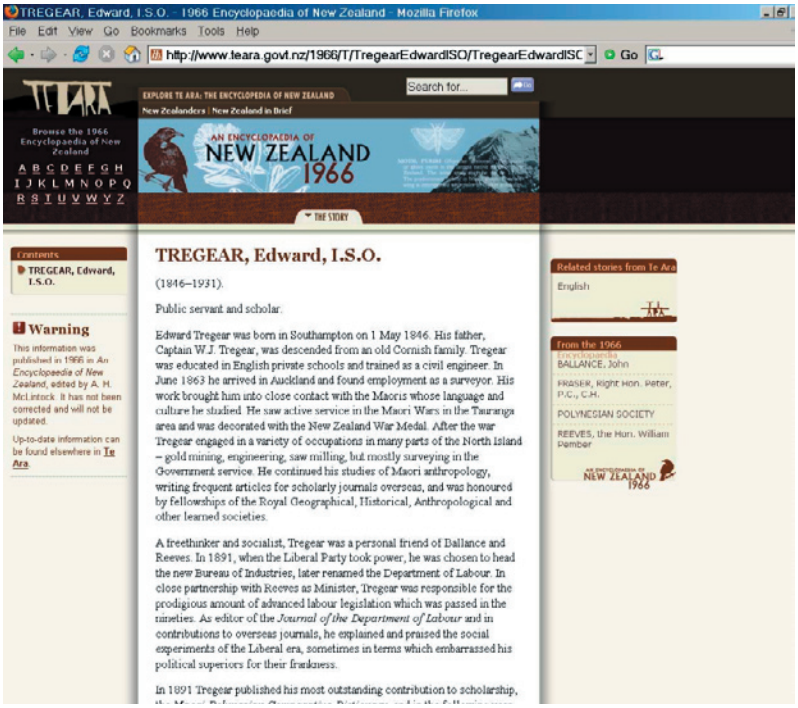
Edward Tregear web page

This page contains links to everything on the site related to Edward Tregear. Note that the *Maori-Polynesian Comparative Dictionary* is listed as one of his works.
Note also the list of works where Edward Tregear is mentioned. Clicking on one of these links will display the page of the topic which represents the text where he is mentioned.



Rollo Arnold's book *New Zealand's Burning* makes a mention of Edward Tregear

The topic map also defines the content of the list of external links on Tregear's page. Clicking on one of these links loads a page from an external site. This external link was harvested along with the Edward Tregear topic from our name authority file.



Edward Tregear external link

Exploring TEI XML Documents with XQuery

James Cummings, Oxford Text Archive, University of Oxford

Abstract:

This paper commences with a basic introduction and survey of the W3C XML Query Language (XQuery), using example XQueries to convey the basics of the language and the potential of using XML Databases. It looks at the various expressions which constitute an XQuery and the functions, mostly inherited from XPath, which can be used to located and extract data from an XML Database. The paper then continues by looking at using XQuery with TEI P5 XML documents in specific, with the popular native XML database eXist, and introduces additional aspects such as the use of namespaces and some of the useful extensions implemented by eXist. How to construct queries to return TEI documents, and parts of TEI documents, based on specific criteria are demonstrated. Following this, the paper explains step-by-step the creation of a basic XQuery web application for retrieving information from an eXist database using XML in conjunction with the Apache Cocoon web publishing framework.

Introduction

This paper originates from the idea that once you have created some TEI XML you then want to do something with it. The majority of people transform this TEI XML to another format (XML, HTML, PDF, Topic Maps, etc.) for display or for use in another program. Others may use it in a specialist XML program suited to their specific research needs (such as Xaira). But there is another option, one can add your TEI XML to a native XML Database and use XML Query to analyse or extract the information from collections of documents.

The XQuery Language

Why use XQuery instead of XSLT or simple XPath to query XML documents? One reason might be that XQuery is domain-specific rather than a general higher-end solution. Unlike XSLT which was originally conceived to transform XML documents to XSL-FO, XQuery has only one purpose -- to query collections of XML documents in a native XML database. This means that it is optimised for its task. If you want to dynamically and unpredictably query a large number of XML documents, then XQuery should give greater overall performance for the task than, say, passing all the documents through XSLT stylesheets. Moreover, a more abstract form of language may be descriptive, saying **how** it wants the Query to be done, whereas XQuery is declarative, saying **what** you want to do. Although with markup languages themselves it makes more sense to be descriptive, with a query language it is arguably more efficient to be declarative. However, XSLT is more efficient at processing the results of your XQuery to a more convenient display format, such as HTML. In most cases it is useful to use a combination of XQuery and XSLT. XQuery can be used to find the set of results, but then these can be passed through a pipeline of XSLT stylesheets for transformation. This uses both languages for the area which is their greatest strength.

But many people have never used XQuery, so it was felt it might be useful to provide an introduction to it. XQuery is a compact and powerful W3C recommendation, based on XPath, and is quite easy to learn. If one is familiar with other query languages, like SQL which is commonly used to query relational databases, then picking up basic XQuery is quite straightforward. As with any other query language, XQuery has a number of different types of expressions and built-in functions.

XQuery Expressions

There are numerous ways in which to express an XQuery and achieve the results you want. It can be useful to look at the component types of expressions that one is most likely to encounter. These include:

- **Path Expressions**
- **Element Constructors**
- **FLWOR Expressions**
- **List Expressions**
- **Conditional Expressions**
- **Qualified Expressions**
- **Datatype Expressions**

Out of this list, the first three are by and large the most common in simple everyday XQueries.

```
//p[foreign/@xml:lang='lat']

//foreign[@lang='lat']/text()

doc('/db/bible/2John.xml')//text

collection('/db/bible')//text//ab[@type='verse']
```

Path Expressions are formulated in XPath and return a set of XML nodes. It is through XPath that XQuery inherits most of its functions. XQuery adds mechanisms for dealing with collections as a whole or specific documents in a collection.

```
<latin>o tempora o mores</latin>

<latin>{$s}</latin>

<ul>
   <li>item one is {$one}</li>
   <li>item two doesn't exist</li>
</ul>
```

Element constructors are used to return the results of the XQuery as an XML fragment. The result of your XQuery should be well-formed XML. However, it doesn't have to be valid to any particular schema or DTD (unless you want it to be). From a query on TEI XML you could output just exact copies of the nodes you find, or TEI XML in a different form, or HTML, or indeed other formats.

This can also be used to embed an XQuery entirely inside some output, for example inside an HTML file.

- **For - Let - Where - Order – Return**
  - for defines a cursor over an XPath selection
  - let defines a name for the contents of an Xpath
  - where selects from the nodes as in SQL
  - order by sorts the results as in SQL
  - return specifies the XML fragments to be constructed
- **Curly braces are used for grouping, and define the scope of the for clause**
- **This is one of the most common forms of XQuery, and is often used for the equivalent of SQL joins**

FLWOR (pronounced 'flower') expressions are a form of expression which allows you to iterate over a node-set to produce the desired results of that query. The FLWOR expression is the most common of the expressions in XQuery. It is equivalent in many ways to the SQL 'SELECT' statement or an XSLT '<xsl:for-each>' loop. It is named for the 5 clauses which possibly make up its content, these are: *for*, *let*, *where*, *order* and *return*. A FLWOR expression starts with one or more *for* and/or *let* clauses (in any order), followed by optional *where* and *order by* clauses, and finally a single *return* clause. This type of expression can be used for anything from a simple query to complex joins between multiple document repositories.

```
for $vulgateBook in collection('/db/bible')/TEI
```

A *for* clause defines a named cursor over an XPath selection. So you define whether you query may be operating over each of the documents in a collection, or all of the the elements, or for any occurrence of a more specific XPath. The content of this *for* clause can be accessed as a variable in other clauses. The *for* clause does not necessarily have to precede a *let* clause, as long as the *let* clause is not dependent on it in any way. The *for* clause limits the scope of the *let* clauses based on it to the current iteration through the node-set. Curly brackets { and } are used both for grouping and to delimit the scope of the *for* clause. In this case we are querying a database containing a TEI XML version of St Jerome's Latin Vulgate Bible.

```
let $title := $vulgateBook/teiHeader/fileDesc/titleStmt/title[1]
let $greekPhrases := $vulgateBook//foreign[@xml:lang='grc']
```

A *let* clause defines the named variable for the contents of an XPath statement. This is often based on a preceding *for* clause, and this variable can be used in following clauses as a convenient method to reference the XPath in question, usually but not necessarily within the scope of the *for* clause within which it is declared. A *let* clause can precede a *for* clause, but then it is outside the scope of this clause. Thus, while it can be used as a variable inside the clause it will remain the same for each iteration.

```
where count($greekPhrases) > 1
```

An optional *where* clause selects a subset of the nodes available to return. It functions along similar lines as a 'WHERE' clause in SQL. This allows one to iterate over a large range of nodes but only

return those which match a specific boolean condition.

```
order by $title
```

An optional *order by* clause controls the order of the node-set used to construct the XQuery results by a set of sort keys. It is the node-set that is going to be returned, as defined by the scope of the *for* clause which is sorted, not just the already-flattened results. The order can be sorted by the value of an existing variable, whether used in the resulting output or not. This can be modified by whether you want the order to be sorted 'ascending' (the default) or 'descending'.

```
return
<li>{$title}</li>
```

The final FLOWR clause is the *return* clause which one follows with an element constructor to create well-formed XML output. There are multiple methods for creating all the usual XML node types: element, attribute, text, document, comment, and processing-instructions. However, the easiest way to do this in most cases is simply to write out the XML directly. Curly brackets are used here as well to embed any XQuery statements inside the output.

```
for $vulgateBook in collection('/db/bible')/TEI
let $title := $vulgateBook/teiHeader/fileDesc/titleStmt/title[1]
let $greekPhrases := $vulgateBook//foreign[@xml:lang='grc']
where count($greekPhrases) > 1
order by $title
return
<li>{$title}</li>
```

As you can see, if you put all the clauses of the FLOWR statement together you can form a pretty sophisticated query if you need to. Here our *for* clause selects each book of the Bible, for each book we assign an XPath of the first title to a variable called *title*. We also define another variable called *greekPhrases*, and when any particular document has more than one of these, we output the title in an <li> element, ordered by title.

The other XQuery expressions are not used nearly as frequently as a FLWOR expression, or sometimes used as part of one. For example, list expressions are used when an XQuery wants to manipulate a list of values. These have many possible operators, but include manipulation values as part of a constant list, over ranges of integers, of part of an XPath expression, concatenation of multiple lists of values, operations to join/split/examine sets, and the use of various list-related specific functions. When a list is treated as a node-set by XQuery, then the XML nodes are compared on node identity and any duplicates are removed with the order preserved unless otherwise sorted.

- **XQuery expressions manipulate lists of values:**
  - constant lists: (7, 9), integer ranges: i to j
  - XPath expressions, concatenation
  - set operators: | (or union), intersect, except
  - functions: count(), sum(), data(), distinct-values()...
- **When lists are viewed as nodesets:**
  - XML nodes are compared on node identity

```
    - duplicates are removed
    -   the order is preserved
```

XQuery also has the ability to process basic conditional expressions of an if/then/else statement. If some particular condition is met, then something is certainly the case so output one thing, otherwise output something else. While this can be used inside FLOWR expressions to great effect, it is also commonly used in defining user-created functions.

```
<div>
    {
       IF document("test.xml")//title/text()
          ="XQuery Test"
       THEN <p>This is true.</p>
       ELSE <p>This is false.</p>
    }
</div>
```

XQuery also allows qualified expressions, this takes two forms, existential and universal quantification, namely 'some-in-satisfies' or 'every-in-satisfies'.

```
for $b in document("book.xml")//text
where some $p in $b//p satisfies
   (contains($p,"sailing") AND
contains($p,"windsurfing") )
return
$b/ancestor::teiHeader//title[1]
```

These work, especially in *where* clauses as part of a FLOWR statement by providing an easy way to select those nodes where either at least one node in the set satisfies the condition or every node in the set satisfies the condition.

```
for $b in document("book.xml")//text
where every $p in $b//p satisfies
   contains($p,"sailing")
return $b/ancestor::teiHeader//title[1]
```

Another feature of XQuery is its datatype support. With datatype expressions your XQuery has access to the stated datatypes of particular values. This means that you can test whether this value is valid against that schema's datatype. It supports all datatypes from the W3C's XML Schema, including both primitive and complex types. When constructing constant values based on datatypes these can be written as literals, constructor functions or explicit casts. Arbitrary schema documents can be imported into an XQuery to allow validation of node-sets against that schema. An 'instance of' operator allows runtime validation of any value relative to a known datatype or schema, while a 'typeswitch' operator allows branching based on datatypes.

- **XQuery supports all datatypes from XML Schema, both primitive and complex types**
- **Constant values can be written:**

- as literals (like string, integer, float)
- as constructor functions (true(), date("2001-06-07"))
- as explicit casts (cast as xsd:positiveInteger(47))

- **Arbitrary XML Schema documents can be imported into an XQuery**
- **An instance of operator allows runtime validation of any value relative to a datatype or a schema**
- **A typeswitch operator allows branching based on datatypes**

XQuery Functions

XQuery inherits many of its functions from XPath, but also defines some of its own. There are functions to deal with accessing node-sets and their context, manipulating strings and substrings, or working with numeric data, boolean values, dates and time, and sequences. While this list is not meant in any way to be exhaustive, but some of the functions are listed below to give you and idea of the types of in-built functions available to an XQuery, but I'm not going to describe them all here.

Node-set and node context functions include: node-name(), nilled(), string(), data(), base-uri(), document-uri(), default-collation(), static-base-uri(), position(), last(), name(), local-name(), namespace-uri(), lang(), and root()
Sequence-related functions: index-of(), empty(), exists(), distinct-values(), insert-before(), remove(), reverse(), subsequence(), unordered(), zero-or-one(), one-or-more(), exactly-one(), deep-equal(), id(), idref(), doc(), doc-available(), collection()
String and substring functions include: codepoints-to-string(), string-to-codepoints(), compare(), codepoint-equal(), concat(), string-join(), substring(), string-length(), normalize-space(), normalize-unicode(), upper-case(), lower-case(), translate(), encode-for-uri(), iri-to-uri(), escape-html-uri(), contains(), starts-with(), ends-with(), substring-before(), substring-after(), matches(), replace(), and tokenize()
Numeric and boolean functions include: true(), false(), not(), abs(), boolean(), ceiling(), floor(), number(), round(), and round-half-to-even(), count(), avg(), max(), min(), sum()
Dates and time: years-from-duration(), months-from-duration(), months-from-duration(), days-from-duration(), hours-from-duration(), minutes-from-duration(), seconds-from-duration(), year-from-dateTime(), month-from-dateTime(), day-from-dateTime(), hours-from-dateTime(), minutes-from-dateTime(), seconds-from-dateTime(), timezone-from-dateTime(), year-from-date(), month-from-date(), day-from-date(), timezone-from-date(), hours-from-time(), minutes-from-time(), seconds-from-time(), timezone-from-time(), adjust-dateTime-to-timezone(), adjust-date-to-timezone(), adjust-time-to-timezone()

XML Namespaces and XQuery

While there is a lot more to XQuery than these very basic concepts, this should be enough to understand the use of XQuery with TEI documents. However, before moving on to some better examples it might be helpful to have an explanation of the declaration of XML namespaces and their use in XQueries, as well as some of the extensions specific to the eXist native XML database in specific.

- **The Namespace of an element, is the scope within which it is valid.**
- **Elements without Namespaces may collided when we combine bits of multiple documents together. XML Namespaces solve this problem and enable using other schemas within yours.**
  (e.g. SVG or MathML inside TEI)
- **An XML Namespace is identified by a URI reference.**
- **XML Namespaces usually appear as qualified names, which contain a single colon, separating the name into a prefix and a local part. The prefix, which is mapped to a URI reference, selects a namespace.**
  (e.g. tei:teiHeader, svg:line)
- **Child elements inherit the namespace declaration of their ancestor.**

An XML namespace defines the scope in which an element is valid. As elements without namespaces may collide when we combine multiple fragments of documents together, namespaces stop the potential for conflicting element names. Moreover, the use of namespaces enables the possibility of embedding elements from another schema inside your document. An XML namespace is defined by using a URI reference, which may or may not point to anything. And, child elements inherit the namespace of their parents. In XQuery a namespace is defined at the top of the document like this:

```
declare namespace tei="http://www.tei-c.org/ns/1.0";
```

Inside an XQuery, as inside an XML document, elements which are not part of the default namespace are referred to with a namespace prefix. (e.g. tei:title, svg:line, html:object) This prefix must be remembered when constructing XQueries. If you remember that XQuery from earlier, this is what it should have looked like:

```
(: This is a comment :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
for $vulgateBook in collection('/db/bible')/tei:TEI
let $title := $vulgateBook/tei:teiHeader/tei:fileDesc/tei:titleStmt/tei:title[1]
let $greekPhrases := $vulgateBook//tei:foreign[@xml:lang='grc']
where count($greekPhrases) > 1
order by $title
return
<li>{$title}</li>
```

As TEI P5 XML is namespaced, you must declare the TEI namespace when using XQuery to query TEI documents. In addition, any reference to a TEI element must then be prefixed with the namespace prefix you have declared. While this may seem tiresome at first, it becomes second nature quite quickly and the benefits of incorporating XML namespaces are very attractive.

eXist Extensions

The eXist native XML database is a stable, mature, free, and open source native XML database favoured by many who use XQuery. In addition to supporting XQuery, it provides a number of

extensions for use with XQuery that can be quite helpful. These include the introduction of both new functions and new operators, and a few of them are worth highlighting. While the built-in XQuery function doc() is restricted to a single URI as its argument, eXist also provides a function, document(), which can accept multiple URI references as arguments and so allow searching across multiple named documents. Similarly the XQuery function collection() is recursive in also searching sub-collections within the named collection, however eXist provides an xcollection() function which ignores sub-collections if this is desired.

Some of the more important eXist extensions, however, are those that simplify text searching. eXist adds two operators for text searching: '&=' and '|='. These allow easy searching of text nodes where the order and distance of the search terms is not important. '&=' is used to select nodes containing **all** the keywords provided, while '|=' finds nodes which contain **any** of the keywords provided.

- **&= means must have all the words**
- **|= means could have any one of the words**
- **Usage: //tei:ab &= 'corde stulto'**
- **This will find paragraphs containing both the words corde and stulto (in either order), and is easier to type than the equivalent Xpath: //tei:ab[contains(.,'corde') and contains(.,'stulto')]**
- **Proximity: //tei:ab[near(.,'corde stulto',20)]**
- **Stem Matching: //tei:ab &= 'cord* stult*'**

eXist also provides a new function for doing proximity searching. This will search the specified node for the keywords provided, similar to '&=', but it will both pay attention to the order in which the search terms are provided, and allow a distance limit (in words) indicating how close they should be to each other.

Complexity and Native XML Databases

Native XML databases work by indexing the documents when you add them to the collection. This means that the database creates relational tables identifying the location of the various nodes of the document. This provides a reliable and efficient method of being able to query the structure of the document, but flat numbering of the nodes creates a limit in the number of nodes which can be indexed. It is not only the size of an XML document which can lead to problems, but more accurately its complexity. This complexity is determined by both the overall number of nodes and how deeply nested elements are in the hierarchy. This makes it impossible to calculate an actual limit since this will vary from document to document. This was particularly a problem with heavily over-balanced trees such as you often get with TEI documents. However, the native XML database eXist has recently introduced a new 'Dynamic Level Numbering' (DLN) which avoids putting a conceptual limit on the size and complexity of documents to be indexed. This also has other positive side effects, for example fast node-level updates which would mean that re-indexing the whole of the node tree after a major update would not necessarily be required. However, some of the lessons learnt from using eXist prior to the introduction of the DLN scheme are still valuable. Specifically, it still can be quite beneficial to split up extremely large documents by divisions in their overall structure. This can lead to slightly quicker retrieval of document node-sets which appear inside a single fragment.

XQuery Examples

XQuery allows you to build up increasingly complex queries from very simple initial logic. As was mentioned earlier, there is a copy of St Jerome's Latin Vulgate Bible in our the eXist native XML database. I chose this because I am also a medievalist and the Vulgate was such an important text for that period. For our purposes the actual content of these files doesn't really matter, it has a very regular structure of, books of the Bible, which contain chapters, which themselves contain verses. This very regular structure makes it a perfect example for using XQuery to extract data based on the structure of an XML document. For example, if we want to produce a list of the books of the Bible, all we may need is a simple XPath query:

```
(: Vulgate: XPath titles :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
collection('/db/bible')//tei:title
```

This is a simplest kind of XQuery. We have used a comment to remind ourselves that this is a list of titles, we have declared the TEI namespace as being denoted by 'tei' and then, for the '/db/bible' collection, we have returned any TEI <title> elements that we can find. Our result, encapsulated in an eXist default root element, would simple be one title from each of the books of the Bible. This isn't a very specific XQuery, since if the <title> element was used anywhere else in this collection, it would find that as well. I just happen to know it is only ever used in these files for the title of the document. If we rewrote this as a proper XQuery instead of solely an XPath statement, it might look something like this:

```
(: Vulgate: List of titles, one per book :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
for $vulgateBook in collection('/db/bible')/tei:TEI
let $title := $vulgateBook/tei:teiHeader/tei:fileDesc/tei:titleStmt/tei:title[1]
return $title
```

This returns a series of <title> elements such as:

```
<title>The Vulgate Book of Deuteronomy</title>
```

While this is a nice form to have it in for display purposes, let's say we've decided that we don't need this "The Vulgate Book of" in front of each of our titles. We just want the name of the book instead, as that is much more convenient for a list of titles. I happen to know that the name of the book is stored in the 'n' attribute of the <text> element. If we re-write our XQuery to use this instead it would look like:

```
(: Vulgate: List of book names :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
for $vulgateBook in collection('/db/bible')/tei:TEI
let $bookName := data($vulgateBook/tei:text/@n)
return $bookName
```

Notice that we use the data() function to extract the text of the attribute; we don't want to recreate an 'n' attribute in our output. What this returns to us is a list of text book names such as:

```
...
Colossians
Daniel
Deuteronomy
Ecclesiastes
...
```

This isn't very useful for further processing, so let's return each of them instead inside a <title> element.

```
(: Vulgate: List of book names :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
for $vulgateBook in collection('/db/bible')/tei:TEI
let $bookName := data($vulgateBook/tei:text/@n)
return
<title>{$bookName}</title>
```

This again returns us a set of results something more like:

```
...
<title>Colossians</title>
<title>Daniel</title>
<title>Deuteronomy</title>
<title>Ecclesiastes</title>
...
```

But let's say that instead of a simple list of titles, we want to produce a list of books and link to the copy of that book. Even though this it is usually identical to the book name, just to be safe we need to get the contents of the <idno> element, which is known to be the same as the name of the file in which the text is stored and used only once in the document. We could output this as a series of TEI <ref> elements:

```
(: Vulgate: ref elements to vulgate books :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
for $vulgateBook in collection('/db/bible')/tei:TEI
let $bookName := data($vulgateBook/tei:text/@n)
let $idno := data($vulgateBook//tei:idno)
return
<ref target="{$idno}.html">{$bookName}</ref>
```

Notice how we have concatenated the .html extension onto the value of the $idno variable. But let's say we didn't want to have to transform this further, and so wanted to output XHTML list-items instead, with a link inside. This would look more like:

```
(: Vulgate: HTML links to vulgate books :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
```

```
for $vulgateBook in collection('/db/bible')/tei:TEI
let $bookName := data($vulgateBook/tei:text/@n)
let $idno := data($vulgateBook//tei:idno)
return
<li> <a href="{$idno}.html">{$bookName}</a></li>
```

And give us a result containing something like:

```
...
<li><a href="Colossians .html">Colossians</a></li>
<li><a href="Daniel .html">Daniel</a></li>
<li><a href="Deuteronomy .html">Deuteronomy</a></li>
<li><a href="Ecclesiastes .html">Ecclesiastes</a></li>
...
```

This is a good start, we could use this to create an index page of all the books of the Vulgate Bible in our XML database. An index page created with such an XQuery would always be up to date, reflecting the contents of the database, rather than a static page needing to be changed every time you added a new book to the Bible database.

If we wanted to ensure that our list of books was sorted into alphabetical order by the name of each book, then we could add an *order by* clause before returning our results.

```
(: Vulgate: Sorted HTML links to vulgate books :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
for $vulgateBook in collection('/db/bible')/tei:TEI
let $bookName := data($vulgateBook/tei:text/@n)
let $idno := data($vulgateBook//tei:idno)
order by $bookName
return
<li> <a href="{$idno}.html">{$bookName}</a></li>
```

Some of you will have noticed that this is just a set of <li> elements, not really an HTML list. Although we could turn it into one with subsequent processing easily enough, we could also wrap it inside an HTML <ul> element by embedding the XQuery inside this element. For example:

```
(: Vulgate: Sorted bulleted HTML links to vulgate books :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
<ul>{
for $vulgateBook in collection('/db/bible')/tei:TEI
let $bookName := data($vulgateBook/tei:text/@n)
let $idno := data($vulgateBook//tei:idno)
order by $bookName
return
<li> <a href="{$idno}.html">{$bookName}</a></li>
}
</ul>
```

Instead of returning just 78 <li> elements, this returns one <ul> element with 78 li elements inside it.

The clever ones here will point out that this still isn't really a proper HTML document. We could do that by embedding the XQuery not only in the <ul> element, but the entire contents of an HTML file.

```
(: Vulgate: Index page for vulgate books :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Vulgate Index</title>
</head>
<body>
<h1>Books of St Jerome's Vulgate</h1>
<ul>{
for $vulgateBook in collection('/db/bible')/tei:TEI
let $bookName := data($vulgateBook/tei:text/@n)
let $idno := data($vulgateBook//tei:idno)
order by $bookName
return
<li> <a href="{$idno}.html">{$bookName}</a></li>
}
</ul>
</body>
</html>
```

Now let's say that we didn't want all the books of the Bible, but only those which contain specific line numbers. So we want to search our texts rather than just listing them. Let's say, for some reason, we want to find any books of the Bible which have chapter/verse numbers of 6:66. Assuming the rest of the HTML file, our embedded XQuery might look something like:

```
(: Vulgate: Query vulgate books :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
<ul>{
let $vulgateBook := collection('/db/bible')/tei:TEI
for $queryHit in $vulgateBook//tei:ab[@n='6:66']
let $bookName := data($queryHit/ancestor::tei:text/@n)
let $idno := data($queryHit/ancestor::tei:TEI//tei:idno)
let $verseID := data($queryHit/@xml:id)
order by $bookName
return
<li> <a href="{$idno}.html#{$verseID}">{$bookName}</a></li>
}
</ul>
```

Notice that not only have we moved the *for* clause in our relative simple search, but that the definitions of the $bookName and $idno variables have changed since we want to go from our $queryHit, back up the tree to find the book's name and id number. (Though, there are other ways of getting this information, it shows you how easy it is to traverse the tree to other points in the hierarchy.) Also notice that since we are looking at a verse (here stored in an anonymous-block or <ab> element), we have grabbed the xml:id attribute for that element and used this to allow our links to go straight to the desired verse.

Building a Basic Web Application

As you can see, with very minimal effort one can build some basic XQueries which could form the foundation of a straightforward but extemely useful web application. While eXist has an XQuery web interface which you can use to process XQueries and see your results as XML, it makes more sense to use eXist from within another context such as the Apache Cocoon web publishing framework. This is a free and open source system which allows you to entire separate content, processing logic, and presentation, while interacting with databases (including eXist), XML documents, and many other web technologies. I'll briefly explore the possibility of using some of the XQueries we've created to fashion a basic web application.

XQuery Request Parameters and Web Forms

eXist enables you to handle input given as parameters, and these requests can be embedded in the URL and accessed by your XQuery. This enables your XQueries to respond to queries from web forms.

eXist uses separate namespaces for functions for handling, requests, responses, sessions, XPath, XQuery and similar functions. The namespace used for handling eXist's request functions is "http://exist-db.org/xquery/request". For creating an interactive web application, the functions of most interest include:

- **request:get-parameter():** which expects two arguments, the name of the parameter and a default value to return if this parameter has not been set. The function results the values for the parameter in question.
- **request:get-uri():** which returns the URI of the current request.
- **request:create-session():** which can be used to create a new HTTP session.

Our interest is in the first of these. For example, our final XQuery above, instead of only looking for books with chapter/verse number '6:66', we could use this as a default if the user hadn't provided us with a number to use. To do this, we'd have to rewrite our XQuery as:

```
(: Vulgate: Query vulgate books :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
declare namespace request="http://exist-db.org/xquery/request";
<ul>{
let $vulgateBook := collection('/db/bible')/tei:TEI
let $query := request:get-parameter('q', '6:66')
for $queryHit in $vulgateBook//tei:ab[@n=$query]
let $bookName := data($queryHit/ancestor::tei:text/@n)
let $idno := data($queryHit/ancestor::tei:TEI//tei:idno)
let $verseID := data($queryHit/@xml:id)
order by $bookName
return
<li> <a href="{$idno}.html#{$verseID}">{$bookName}</a></li>
}
</ul>
```

Note that we've declared a new namespace 'request', and introduced a new variable $query which has a default value of '6:66'. Instead of specifying that this is the value we want in constructing our $queryHit variable, we refer back to the content of this parameter when we need it. Otherwise the XQuery is the same as earlier.

Apache Cocoon Sitemap

Apache's Cocoon web publishing framework uses a 'sitemap' file to define what happens as result of specific URLs within its scope.

```
<map:match pattern=" SOME PATTERN ">
  <map:generate src=" SOME LOCAL OR REMOTE SOURCE"/>
  <map:transform src=" SOME XSLT STYLESHEET"/>
  <map:transform src=" ANOTHER XSLT STYLESHEET"/>
  <map:transform src=" XHTML TEXT PDF etc."/>
</map:match>
```

This file, itself in XML, allows a pipelining of XML transformations, whether originating from an eXist database, the file system, and whether using multiple XQueries or XSLT transformations to produce the end result. This allows a modularity, a separation of transformations into steps to allow greater ease of control over any individual step. A basic sitemap might contain a number of pipeline statements.

Example Web Application

```
<!-- Vulgate Index Page -->
<map:match pattern="index.html">
   <map:generate src="index.xq" type="xquery"/>
   <map:serialize type="xhtml"/>
</map:match>

<!-- Any Other Vulgate HTML File -->
<map:match pattern="*.html">
   <map:generate src="xmldb:exist:///db/bible/{1}.xml"/>
   <map:transform src="vulgate2html.xsl"/>
   <map:serialize type="xhtml"/>
</map:match>

<!-- View Vulgate file as XML -->
<map:match pattern="*.xml">
   <map:generate src="xmldb:exist:///db/bible/{1}.xml"/>
   <map:serialize type="xml"/>
</map:match>
```

Here part of our basic Vulgate Bible sitemap gives us 3 URL patterns to match. The first says if we are asked for a file called index.html then use an xquery called index.xq to generate a list of all the

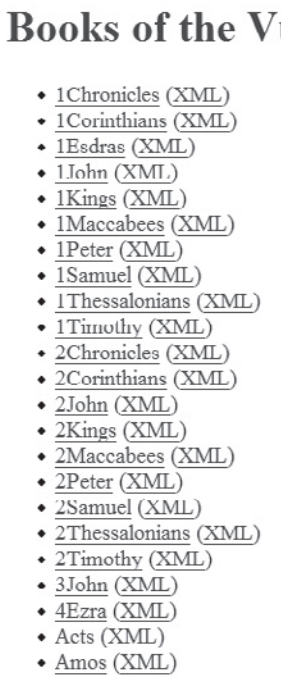
book titles and serialize the result as XHTML. The second is less specific and says if we are asked for anything (asterisk is a wildcard) ending in '.html' to go and get the corresponding XML file from our eXist database, and pass it through a vulgate2html XSLT stylesheet to turn it into XHTML. The third says if we are asked for the XML source of anything, then just get it from the database and just give it out as XML.

If we wanted to use an XQuery to search our texts, then adding a sitemap entry for that, which here also tells cocoon to pass on any parameters it gets to the xquery (though this can be set more globally) is quite straightforward.

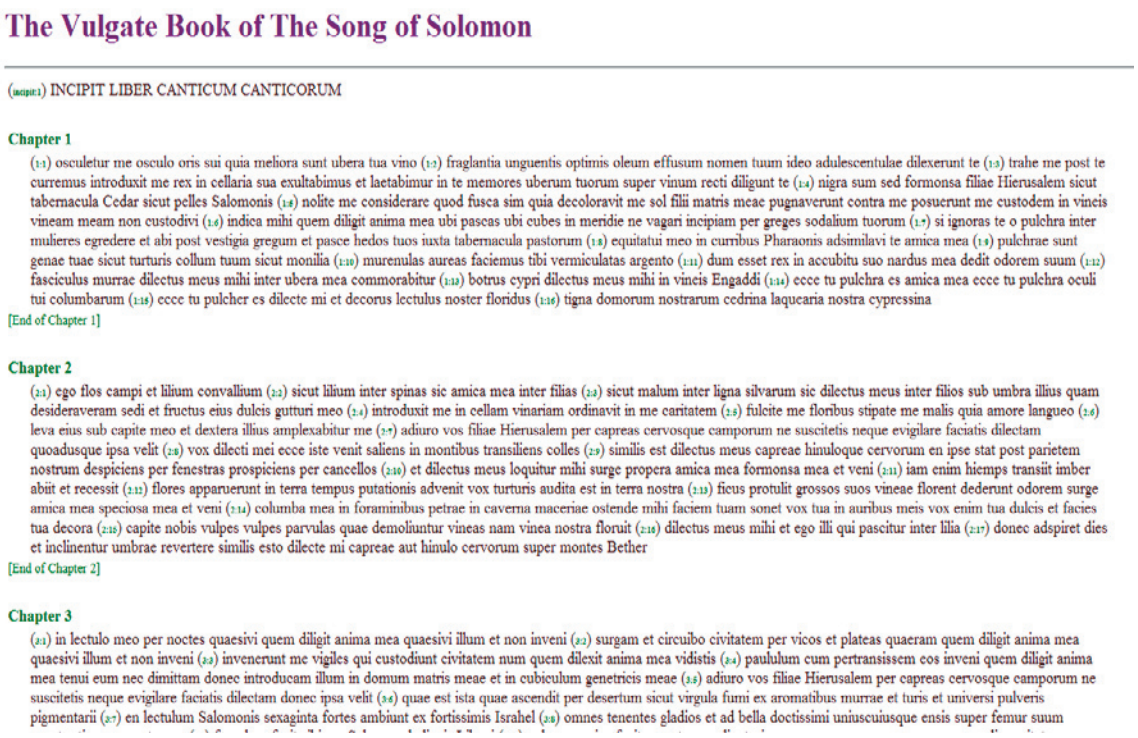
```
<!-- Any Other Vulgate HTML File -->
<map:match pattern="search.html">
   <map:generate src="search.xq">
         <map:parameter name="use-request-parameters" value="true"/>
         <map:parameter name="create-session" value="true"/>
   </map:generate>
   <map:transform src="searchResult2html.xsl"/>
   <map:serialize type="xhtml"/>
</map:match>
```

Result

And this is the result from the XQuery which generates a list of titles starting as below:



And the individual books are grabbed from the XML Database and transformed into XHTML for display by a separate brief XSLT stylesheet.



The Vulgate Book of The Song of Solomon

(incipit) INCIPIT LIBER CANTICUM CANTICORUM

Chapter 1

(1:1) osculetur me osculo oris sui quia meliora sunt ubera tua vino (1:2) fraglantia unguentis optimis oleum effusum nomen tuum ideo adulescentulae dilexerunt te (1:3) trahe me post te curremus introduxit me rex in cellaria sua exultabimus et laetabimur in te memores uberum tuorum super vinum recti diligunt te (1:4) nigra sum sed formonsa filiae Hierusalem sicut tabernacula Cedar sicut pelles Salomonis (1:5) nolite me considerare quod fusca sim quia decoloravit me sol filii matris meae pugnaverunt contra me posuerunt me custodem in vineis vineam meam non custodivi (1:6) indica mihi quem diligit anima mea ubi pascas ubi cubes in meridie ne vagari incipiam per greges sodalium tuorum (1:7) si ignoras te o pulchra inter mulieres egredere et abi post vestigia gregum et pasce hedos tuos iuxta tabernacula pastorum (1:8) equitatui meo in curribus Pharaonis adsimilavi te amica mea (1:9) pulchrae sunt genae tuae sicut turturis collum tuum sicut monilia (1:10) murenulas aureas faciemus tibi vermiculatas argento (1:11) dum esset rex in accubitu suo nardus mea dedit odorem suum (1:12) fasciculus murrae dilectus meus mihi inter ubera mea commorabitur (1:13) botrus cypri dilectus meus mihi in vineis Engaddi (1:14) ecce tu pulchra es amica mea ecce tu pulchra oculi tui columbarum (1:15) ecce tu pulcher es dilecte mi et decorus lectulus noster floridus (1:16) tigna domorum nostrarum cedrina laquearia nostra cypressina

[End of Chapter 1]

Chapter 2

(2:1) ego flos campi et lilium convallium (2:2) sicut lilium inter spinas sic amica mea inter filias (2:3) sicut malum inter ligna silvarum sic dilectus meus inter filios sub umbra illius quam desideraveram sedi et fructus eius dulcis gutturi meo (2:4) introduxit me in cellam vinariam ordinavit in me caritatem (2:5) fulcite me floribus stipate me malis quia amore langueo (2:6) leva eius sub capite meo et dextera illius amplexabitur me (2:7) adiuro vos filiae Hierusalem per capreas cervosque camporum ne suscitetis neque evigilare faciatis dilectam quoadusque ipsa velit (2:8) vox dilecti mei ecce iste venit saliens in montibus transiliens colles (2:9) similis est dilectus meus capreae hinuloque cervorum en ipse stat post parietem nostrum despiciens per fenestras prospiciens per cancellos (2:10) et dilectus meus loquitur mihi surge propera amica mea formonsa mea et veni (2:11) iam enim hiemps transiit imber abiit et recessit (2:12) flores apparuerunt in terra tempus putationis advenit vox turturis audita est in terra nostra (2:13) ficus protulit grossos suos vineae florent dederunt odorem surge amica mea speciosa mea et veni (2:14) columba mea in foraminibus petrae in caverna maceriae ostende mihi faciem tuam sonet vox tua in auribus meis vox enim tua dulcis et facies tua decora (2:15) capite nobis vulpes vulpes parvulas quae demoliuntur vineas nam vinea nostra floruit (2:16) dilectus meus mihi et ego illi qui pascitur inter lilia (2:17) donec adspiret dies et inclinentur umbrae revertere similis esto dilecte mi capreae aut hinulo cervorum super montes Bether

[End of Chapter 2]

Chapter 3

(3:1) in lectulo meo per noctes quaesivi quem diligit anima mea quaesivi illum et non inveni (3:2) surgam et circuibo civitatem per vicos et plateas quaeram quem diligit anima mea quaesivi illum et non inveni (3:3) invenerunt me vigiles qui custodiunt civitatem num quem dilexit anima mea vidistis (3:4) paululum cum pertransissem eos inveni quem diligit anima mea tenui eum nec dimittam donec introducam illum in domum matris meae et in cubiculum genetricis meae (3:5) adiuro vos filiae Hierusalem per capreas cervosque camporum ne suscitetis neque evigilare faciatis dilectam donec ipsa velit (3:6) quae est ista quae ascendit per desertum sicut virgula fumi ex aromatibus murrae et turis et universi pulveris pigmentarii (3:7) en lectulum Salomonis sexaginta fortes ambiunt ex fortissimis Israhel (3:8) omnes tenentes gladios et ad bella doctissimi uniuscuiusque ensis super femur suum

Conclusion

Although none of the examples here have been very sophisticated they demonstrate the basic building blocks which can be used to create a much more intricate XQuery-powered web application. A few basic XQuery scripts, with some very basic XSLT to dynamically transform the results to HTML, and a very minimal Cocoon setup is all that is needed to allow the display of XML files, presentation of search results, and even a consistent web front-end. It is hoped that the basic introductions here have suggested some of the greater possibilities that the combination of these technologies could bring to the content you wish to interrogate. Thank you.

Further Reading

- XQuery specification: http://www.w3.org/XML/Query/
- XSL family of specifications: http://www.w3.org/Style/XSL/
- eXist native XML database: http://www.exist-db.org/

- Text Encoding Initiative: http://www.tei-c.org/
- Apache Cocoon: http://cocoon.apache.org/
- W3Schools XQuery tutorial: http://www.w3schools.com/xquery/default.asp
- XQuery resources: http://www.xquery.com/
- XQuery mailing list: http://xquery.com/mailman/listinfo/talk
- Brundage, Michael, *XQuery: The XML Query Language*, (Addison Wesley, 2004), 505 pages, ISBN: 0321165810
- Katz, Howard (ed.), *XQuery from the Experts: A Guide to the W3C XML Query Language*, (Addison Wesley, 2003), 512 pages, ISBN: 0321180607
- Walmsley, Priscilla *XQuery* (Oreilly, 2006), 600 pages, ISBN: 0-596-52788-8

TEI Day in Kyoto 2006: Abstracts

Paper presentations

Why was and is TEI unknown in Japan and will it become better known?

TUTIYA Syun (Chiba University)

Japan did send delegates to the first preparatory meeting of the TEI and maintained interest in the developments up to P2. Beyond that, the interest remained at the personal level with the result that researchers within the Humanities with a knowledge even of the existence of TEI are extremely rare. The reason for this is not simply that the interest has been low, it also brings to light the fact that researchers in the disciplines of Humanities and Social Sciences do have a concept of textual documents that is very much orientated towards the use of a text. In this paper, I will attempt to investigate the concept of textual documents, the actual use of texts and its change in the 1990s in Japan, and will illuminate changes that have occurred or not occurred in some disciplines. Finally, I will try to highlight what problems this articulates for the preservation in electronic form of textual documents (in a very broad sense) in Japan and propose some steps towards tackling this problem.

Languages with scarce textual materials and markup technologies

MATSUMURA Kazuto (University of Tokyo)

Spoken words are inherently transient; unless recorded to a tape or written down with some kind of writing system, they will soon be gone, sucked without trace into the spatial-temporal past. It is said that the number of languages spoken on this planet is about 6900, most of these are only used in spoken form, so-called 'minority languages without writing system', if the speakers of these languages disappear, so will their language disappear without trace from the face of the earth.

On the other side, languages without writing system are precious ressources for linguists, therefore there are quite a few examples of languages for which linguists did create a written notation using a phonetic transcription. To make these valuable ressources available for computer-based processing, the digitalization and markup of them is an important task for linguists. This presentation will present some of the experiences and lessons learned.

Marking up spoken dialog corpora

TUTIYA Syun (Chiba University), ITAHASHI Shuichi (National Institute of Advanced Industrial Science and Technology, National Institute of Informatics), OHSUGA Tomoko (National Institute of Informatics)

The recording of spoken dialogs is interesting also since it does away with the intrinsic linearity of language. The mechanism that is necessary to explore this within the framework of a TEI document, together with real-world encoding examples will be presented. 128 recordings of spoken dialog that had been made at Chiba University in 1993 and for which the dialogs along with information about the participants and the length of the utterances have been transcribed, this will be used to illustrate the type of problems that had been encountered in this work. We will also discuss a model for representing the phenomena of spoken language that is the prerequisite for such a transcription and will show how this can be realized in a conforming TEI document.

Markup problems: Syntactical analysis and steps to their resolution

OHYA Kazushi (Tsurumi University)

In the process of digitalizing textual ressources used in the humanities and subsequently adding markup to them, there are, I think, mainly three reasons that difficulties are encountered:

1. the source material does not have been sufficiently analyzed
2. markup as a method is at odds with the original aims
3. markup technologies as such have not been sufficuently mastered.

However, the last point also includes the fact that markup technologies themselves have not yet matured sufficiently, so that because of the way the XML standard is defined, some needed constructs can not be written easily. Among other things for example, although the combination of several XML applications (among them for example TEI) is difficult to achieve because of the way the standard is defined, there have nevertheless a large number of applications been defined and widely used. This combination of several applications is something even specialists in markup languages achieve only with difficulties. Not just data input or data conversion, but decide how to encode what is indeed a task that requires high level skills.

On the other hand, adding markup itself is something that is very close to home for a scholar trained in the humanities. In this paper, I will focus on one of the above difficulties, that is the "syntactically induced" problems and pitfalls of markup languages. This is not because of underspecification of the standard, but rather is a consequence of the inherent freedom of description in a markup system. In the application of TEI, some problems encountered are due to the way markup languages as such are defined, others result from the specific text type used. This differentiation is helpful in understanding the way such problems are handled within the TEI, but they can also be applied to markup languages in general. Markup languages are not something that should simply be used since it is defined as a standard, but since they use formal languages to apply annotations as a description on a meta-level, they provide a means to analyze and reflect upon this act as such.

TEI: an Overview

Syd Bauman (Brown University), Lou Burnard (Oxford University)

This talk will present a broad overview of the TEI Consortium and the TEI Guidelines (P5).

The Text Encoding Initiative (TEI) Guidelines have become one of the central tools of the digital humanities landscape, and have revolutionized the creation and use of digital texts for scholarly research. Produced and maintained by the TEI Consortium, the Guidelines are now used widely in a range of scholarly applications, including digital libraries, scholarly editions, manuscript and historical archives, linguistic corpora, individual scholarly projects, and thematic research collections. The TEI community has produced an immensely useful text encoding system that works on many levels — for both simple and very complex forms of data representation — to ensure that humanities texts can be created, stored, exchanged, and archived in a manner that is both effective and expressive. This talk will first describe the TEI Guidelines as a text encoding standard, and their diversity of use within the community of TEI projects. Different disciplinary communities have produced their own specifications for using the TEI Guidelines, and these will be briefly discussed with respect to how they relate to the TEI Guidelines themselves. I will then present how the Guidelines themselves are technically organized, with an overview of how customizations for use by individual projects are produced. I will breifly discuss how the TEI Consortium itself is organized, with an emphasis on the work of the TEI Special Interest Groups, which reflect particular community interests and research areas. In concluding, I will describe the various ways in which individuals, projects, and institutions may become more closely involved.

Towards an internationalized and localized TEI

Sebastian Rahtz (Oxford University)

The Text Encoding Initiative Guidelines have been widely adopted by projects and institutions in many countries in Europe, the Americas, and Asia, and are used for encoding texts in dozens of languages. However, the Guidelines are written in English, the examples are largely drawn from English literature, and even the names of the elements are abbreviated English words. We need to make sure that the TEI and its Guidelines are **internationalized** and **localized** so that they are accessible in all parts of the world.

The paper describes how the TEI project can develop internationally, including

- A review of why localisation and internationalisation matter
- A discussion of how the TEI architecture can be leveraged to support internationalised versions
- The application of the W3C ITS guidelines to the TEI work
- Practical results from a pilot project, and future translation plans
- The tools needed to make use of an internationalised TEI
- The steps towards ontologies in the TEI

XML mark-up of biographical and prosopographical data

Matthew J. Driscoll (Kopenhagen University)

My paper will present work currently under way within the TEI for marking-up biographical and prosopographical data, in other words information on people, including such things as dates and places of birth and death, marriage and family relations, social origins, places of residence, education, occupation, religion, experience of office and so on.

Exploring TEI XML documents with XQuery

James Cummings (Oxford Text Archive)

This paper commences with a basic introduction and survey of the W3C XML Query Language (XQuery), using example XQueries to convey the basics of the language and the potential of using XML Databases. It looks at the various expressions which constitute an XQuery and the functions, mostly inherited from XPath, which can be used to located and extract data from an XML Database. The paper then continues by looking at using XQuery with TEI P5 XML documents in specific, with the popular native XML database eXist, and introduces additional aspects such as the use of namespaces and some of the useful extensions implemented by eXist. How to construct queries to return TEI documents, and parts of TEI documents, based on specific criteria are demonstrated. Following this, the paper explains step-by-step the creation of a basic XQuery web application for retrieving information from an eXist database using XML in conjunction with the Apache Cocoon web publishing framework.

Presenting TEI texts using topic maps

Conal Tuohy (New Zealand Electronic Text Centre, Victoria University)

This presentation is about a method for presenting complex TEI texts.

Many electronic text archives transform their TEI texts into HTML for publishing their texts on the World Wide Web. Typically each chapter or page is transformed from TEI into a separate web page. Such a method produces websites that have the same structure as a physical book.

However, TEI is more powerful than HTML and can encode many other features of interest than just chapters, pages, and paragraphs. For example, TEI is also used to encode information about people and places and events, as well as literary criticism, and linguistic analysis. Indeed, TEI is designed to be extended to suit all kinds of scholarly needs.

These more complex aspects of text encoding are more difficult to transform into HTML. Because TEI is designed to be convenient for scholars to encode complex information, rather than for readers to understand it, it is necessary to transform the TEI into another form suitable for display. For instance, where a TEI corpus includes references to people, these references might be collated together to produce an index. For practical purposes, it is often necessary to extract information from TEI into a database, so that it can be queried conveniently and transformed into a web site.

The new "Topic Map" standard of the International Standards Organisation is a suitable technology for solving this problem. A topic map is a kind of Web database with an extremely flexible structure. This presentation will demonstrate and describe a framework for using TEI together with Topic Maps to produce large websites which can be navigated easily in many directions.

Poster presentations

TEI @ RCH

Dot Porter (Collaboratory for Research in Computing for Humanities, University of Kentucky)

The Collaboratory for Research in Computing for Humanities uses TEI P5 in all its new and developing projects. Our poster will highlight current and developing projects in RCH, and the various ways that we are taking advantage of the flexibility offered by the TEI P5.

Through the Neolatin Colloquia Project, directed by Ross Scaife, Professor of Classics, graduate students and faculty associated with the UK Institute for Latin Studies are creating a variety of materials for the renewed study and enjoyment of neo-Latin colloquia scholastica, texts that date primarily from the 16th and 17th centuries. Modules used to encode the colloquia include those for Performance Texts (drama), Names and Dates (namesdates), and Common Core (core) – especially for the tagging of bibliographic citations and references.

The Latin Lexicography Project (LLP), also directed by Scaife, is building a web-accessible Latin dictionary, initially populated by digitizing and harmonizing the markup of several important Latin lexica with coverage up to about 1850 CE, then growing ever more comprehensive through the assimilation of additional lexica. For the LLC, we are using the dictionaries module to progressively mark up a number of classical Latin and neolatin dictionaries originally published in print.

Still in the planning stages, the Collectio Dacheriana Project directed by Abigail Firey, History Department, will make extensive use of the Critical Apparatus tags (in the textcrit module) in order to record the many variants in this collection of Carolingian canon law.

Under the direction of Ben Withers, Associate Professor and Chair of the Art and Art History Department, the Old English Hexateuch Project will bring together a group of Anglo-Saxon scholars with a variety of specialties to build an edition of the extensively illustrated tenth-century manuscript, British Library Claudius B iv. The edition will make extensive use of the Manuscript Description module (msdescription), and will also propose extensions to the modules for Text Criticism (textcrit) and Transcription of Primary Sources (transcr).

Venetus A Project, in cooperation with Harvard's Center for Hellenic Studies, is part of the Homer Multitext Project. The Venetus A project seeks to create a complete image-based edition of the Biblioteca Nazionale Marciana, Venice, Venetus A, a tenth-century Byzantine manuscript containing the earliest copy of Homer's Iliad plus several layers of annotations. Venetus A will take advantage of the Manuscript Description module (msdescription), and in addition will illustrate TEI interaction with the Classical Text Services (CTS) protocol, and image-text mapping between TEI and METS.

The Versioning Machine

Susan Schreibman (University of Maryland)

The Versioning Machine is open source software for displaying and comparing multiple versions of texts. The display environment provides for features traditionally found in codex-based critical editions, such as annotation and introductory material, while taking advantage of opportunities of electronic publishing, such as providing a frame to compare diplomatic versions of witnesses side by side, allowing for manipulatable images of the witness to be viewed alongside the diplomatic edition, and providing users with an enhanced typology of notes.

The Versioning Machine supports display of XML texts encoded according to the guidelines of the Text Encoding Initiative (TEI). Texts may be encoded individually (as separate documents) or may be encoded according to the TEI's "critical apparatus tagset" (TEI.textcrit) to encode all witnesses in one XML file. The critical apparatus tagset offers the most efficient and thorough methodology for inscribing variants in a structured, machine-readable format. The Versioning Machine provides for enhanced functionality of texts encoded according to this tagset via synchronized scrolling and line matching.

This poster session will demonstrate the use of and applications of "The Versioning Machine".

Using the TEI gaiji module

Christian Wittern (Kyoto University)

The TEI working group on character encoding has developed a module that allows the representation of characters beyond those encoded in the Unicode standard, which is the character encoding standard used in XML and therefore also in TEI. This module adresses to types of problems:

- for a character that *has been specified* in Unicode, the encoder whishes to describe specificly what glyph has been used to render this character, out of the many possibilites that are used to represent it
- for a character that exists *not (yet?)* in Unicode, the encoder needs to represent it somehow

In this poster, some examples are given to show the applications and usages of this module.

The CBETA electronic Tripitaka: An example of a succesful application of TEI to a large premodern Chinese text corpus

Christian Wittern (Chinese Electronic Buddhist Text Association)

The Chinese Electronic Buddhist Text Association (CBETA) embarked on an ambitions project to digitize the whole of the Chinese Buddhist Tripitaka in 1998. With the help of a small, engaged team and largely through efficient use of text-processing and markup technologies, within the last eight years, a total of 100 Volumes amounting to more than 120 million characters have been distributed free of charge over the Internet and on CD-

ROM in a large variety of formats, reaching from PDF files ready to print to text files that are formatted to be read on hand-held devices or cell-phones.

Considerable effort has been made to not only produce a highly accurate transcription of the source text, but also to correct numerous misprints and to add textcritical notes, thus making it a ressource highly praised by scholars of Buddhism all over the world. Technically, a highly customized version of TEI P4 has been used internally, but more recently an experimental version based on TEI P5 has been made available. Taking advantage of the P5 gaiji module, this version now encodes all the more than 8000 characters or variant forms used in these texts, but not yet found in Unicode in a standardized and exchangeable form.

In this poster an application for reading and studying these texts will be showcased.

Navigating a Sea of Texts: Topic Maps and the Poetry of Algernon Charles Swinbure.

John Walsh and Michelle Dalmau (Indiana University)

Topic Maps, including their XML representation, XML Topic Maps (XTM), are powerful and flexible metadata formats that have the potential to transform digital resource interfaces and support new discovery mechanisms for humanities data sources, such as large collections of TEI-encoded literary texts. Proponents of topic maps assert that topic map structures significantly improve information retrieval, but few user-based investigations have been conducted to uncover how humanities researchers and students truly benefit from the rich and flexible conceptual relationships that comprise topic maps.

The proposed poster will provide an introduction to Topic Maps and how a collection of TEI-encoded literary texts, specifically, the Swinburne Project http://swinburnearchive.indiana.edu, benefit from the use of topic maps. The poster will also provide an overview of the methodology used for the comparative usability study that was designed to assess the strengths and weaknesses of a topic map-driven interface versus a standard search interface. The interfaces that were presented to users will be demonstrated along with key findings from the usability study. Lastly, design alternatives based on the usability findings will also be presented.

The results of this study are intended to move the discussion of topic maps in the digital humanities beyond demonstrating the novel to providing evidence of the impact of Topic Maps and their extension of existing classificatory structures on the humanities researcher's discovery experience. We hope to provide those who are implementing topic maps or similar metadata structures in digital humanities resources with design recommendations that will ensure successful user interaction.

Untangling Āgama literature - A Digital Comparative Edition of the Bieyi za ahan jing

Marcus Bingenheimer (Chung-Hwa Institute of Buddhist Studies)

The Digital Comparative Edition of the Bieyi za ahan jing is a project undertaken by the the Chung-hwa Institute for Buddhist Studies, Taipei (www.chibs.edu.tw) and funded by a three- year grant from the Chiang Ching-kuo Foundation for Scholarly Exchange 蔣經國基金會 (www.cckf.org/index-e.htm).

The Bieyi za ahan jing 別譯雜阿含經 (BZA) in 16 fascicles containing 364 sutras belongs to the early Chinese Buddhist texts collectively called Ahan (Āgama) sutras 阿含經. Ahan literature constitues the earliest stratum of Buddhist literature. The originals (in Buddhist Sanskrit) are largely lost, only a few fragments have survived. Next to the Chinese tradition only the Theravāda tradition has preserved a comprehensive set of these sutras in Pāli. While the Nikāyas, as the Ahan sutras are called here, have been extensively studied and fully translated into English, Japanese and German, there are extremely few translations or critical editions of the Chinese Ahan sutras.

Generally, all of the 364 short sutras contained the BZA have at least one parallel in Chinese and one Pāli parallel (with commentary). Often there are several parallels in Chinese and Pāli, sometimes even a fragment in Buddhist Sanskrit has survived.

The aim of the project is to create a digital comparative edition of the BZA, which clarifies these text-clusters. The edition will be freely available to the public. Moreover we are working on an English translation of the BZA text. Textbase for Chinese is the CBETA edition, for Pāli text the Vipassana Research Institute has granted us permission to use the text of the Chaṭṭha Saṅgāyana CD.

The markup of the XML files is designed according to the encoding scheme of the Text Encoding Initiative (TEI) which is transformed into HTML for the user. The markup expresses the basic dialogic structure of the content, names, differentiates between prose and verse parts, and connects them to the authoritative printed versions. For the Pāli and longer Chinese parallels the markup distinguishes between larger parallel and non-parallel passages.

The texts within a cluster are linked through a comparative catalog. If time allows, we will add phrase-level markup for better alignment of the parallels within a text-cluster. Middleware between the source files and the user application will be eXist, an XML database. The delivery system based on eXist is a first for Buddhist Studies as well as Humanities Computing in Taiwan. The end-user selects the cluster s/he wants to view online and can further select which of the texts in the cluster to display, provisionally in a three column layout.

The comparative digital edition:

- enables the user to conveniently compare the different texts of a cluster
- refines and expands the contents of the 364 clusters
- adds a new punctuation to the BZA and the ZA sutras
- provides an annotated English translation of selected sections of the BZA
- enables statistical analysis by creating parallel corpora
- is extensible and allows for further material to be added
- serves as model for future projects that try to reorganize and represent the maze of Āgama literature

XXQ: a query language for XML corpora

Lou Burnard (Oxford University)

This poster will intruduce XXQ -- the new XML query language currently under development for use with XAIRA (XML Aware Indexing and Retrieval Architecture). The poster will also introduce Xaira, of course, but the main focus will be on the idea of an engine-independent query language for XML text. We maybe could have called it Xpath-plus, but the key features about it are

- (a) it's not Xquery so it doesn't try to pretend XML structures are relations
- (b) it's not Xpath so it doesn't restrict you to searching within a single hierarchy
- (c) it's not grep so its atoms are lexical tokens rather than characters.

It is a pattern matching language with the expressive power of regular expressions, and comparable weaknesses (no look-ahead), but one which is represented by a simple XML vocabulary.

Markup problems: Syntactical analysis and steps to their resolution

OHYA Kazushi (Tsurumi University)

One of the difficulties in applying markup is often perceived as result of the syntax of markup languages. In this poster, I will present some of the frequently encountered problems. This is intended primarily for those beginning their work with markup and intend to prepare data according to the TEI guidelines.

Japanese translation project of the TEI Guidelines

OHYA Kazushi (Tsurumi University), Christian Wittern (Kyoto University)

In order to provide support to the encoding of texts in Japanese, a project to translate the TEI Guidelines into Japanese has started. At the moment, a rough draft version of large parts of the P4 version has been prepared. This will be enhanced and refined, with the ultimate goal of preparing a Japanese version of P5, as soon as that becomes stable enough to be translated. This poster will present the current state of the work and hopes to attract more collaborators in this project.

Markup of the "Comprehensive Mirror for Aid in Government"

NAKADATE Hamana (Kyoto University)

The research group responsible for the "Construction of a knowledgebase of Chinese-character-based documents" within the 21st Century COE program at Kyoto University "Toward an Overall Inheritance and Development of Kanji Culture" commenced work on markup of the section pertaining to the Tang period (618-906) of the "Comprehensive Mirror for Aid in Government" (Zizhi tongjian), a well known history compiled by Sima Guang in the mid of the eleventh century. Aim of this work is to annotate names of persons, places and works, extract and expand this information into a comprehensive networked ressource and thus combine traditional textual and historical scholarship with the digital technology of the 21st century and try to lay a basis for new developments within East Asian Studies. In this poster, the state of the work, some examples, problems and possibilities will be shown.

国際セミナー **TEI Day in Kyoto 2006:** アブストラクト集

研究報告

TEI はなぜ日本で知られなかった、知られていないか、知られるようになるか

土屋俊 **(**千葉大学**)**

日本は、その第 1 回準備会議から参加者を派遣し、P2 段階までも関与をつづけ てきたが、それ以降は個人的関与にとどまり、TEI について知る人文社会系研究者は現在もきわめて少ない。この原因はたんに関心の程度が低かったからではなく、日本の人文社会科学研究がもつ文献概念、文献利用の実態によるものであることを示す。これに基づいて、そのような文献概念、文献利用の実態が 1990 年以降どのように変化しているか、そして領域によっては変化していなか を明らかにして、日本の(広い意味での)文献を電子化して保存することに対する課題を明確にし、あわせてその解決方法を提案する。

文字化された言語資源の少ない言語とテクストのマークアップ

松村一登 **(**東京大学**)**

話されたことばは，一過性のものであり，録音や文字表記の形で記録しておかない限り， 発話されるやいなや，たちまち過去という時空に吸い込まれ消えてしまう。現在，この地 球上で使われている言語の数は，およそ６９００と言われるが，その大部分は，話しことばとしてのみ使われている，いわゆる「文字のない少数言語」で，もし話し手がひとり もいなくなれば，痕跡を残さずにこの地上から消えてしまう。他方，文字のない言語の言 語資料が，言語学者たちによって，音声表記を使って，文字化されていることも決して珍 しくはない。この貴重な言語資源をコンピュータ処理できるように，電子化し，マーク アップすることが，言語学者たちの重要な課題となりつつある。

音声対話コーパスのマークアップ

土屋俊 (千葉大学) 、板橋秀一 (産業技術総合研究所、国立情報学研究所) 、
大須賀智子 (国立情報学研究所)

音声対話の記録は、言語がもつ線状的な特性を破壊する点で興味深いものであるが、それを TEI 文書として記述するために必要なメカニズムを実際のコーディ ング例に即して検討する。1993 年に千葉大学で収録された音声対話 128 件について、発話者、発話時間などを含んだ文書を作成する際に問題となった点を議論するとともに、音声言語の記録の前提となる音声言語現象の一般的な形式的 モデルについて提案を行ない、それらのすべてに整合的な TEI 文書を実現する ことが可能であることを示す。

マークアップの課題を **syntax** から見た分類と解決のステップ

大矢一志 (鶴見大学)

人文科学研究で使われる資料を電子化し、それにマークアップ(markup)を施す際に困難を感じる原因は、主に 3 つ、1)対象データの分析が十分でない、2) マークアップという手段が本来の目的と合わない、3)マークアップ技術の理解 が十分ではない、ことが考えられる。但し、3)にあるマークアップ技術は、まだ十分に成熟したものではなく、そのため、例えば XML といった規格自体が持つ不備が原因で「上手く書けない」ことがある。特に、XML は、アプリケーショ ン(e.g. TEI もそのひとつ)を複数関連づけることが規格上困難であるにも関わらず、多くのアプリケーションが提案され、利用されている。実は、複数の XML アプリケーションを関連づけ、統合する方法は、ML(markup languages)の専門家でも解決策は一意に定まらない。単なるデータ入力や変換をするのでは なく、はじめからどう書く(マークアップ)すべきかを決めることは、かなり高 度な作業になっている。

しかし、マークアップすること自体は、人文科学研究者にとってはとても身近 な行為である。本稿では、マークアップする際に困難と感じる原因のうち、ML の Syntax から見た「ひっかかりどころ」を紹介し、規格の不備に惑わされることなく、マークアップが本来持つ自由な記述を再確認したい。これは、TEI を 利用する際、「ML 一般」と「個別テキストタイプ」という 2 つの問題を扱う TEI の論議を、整理して読み進めるヒントとして有効だろう。さらに、ML 一般の問 題を検討する際の、手助けになるかもしれない。ML は、単に利用される規格と してあるだけではなく、従来、ひとがアノテーションとしてきた記述の行為が、 形式言語の側面からメタ記述の行為として評価されうる可能性を探る糸口になっている。

TEI 概説

Syd Bauman (ブラウン大学)、Lou Burnard (オックスフォード大学)

本発表では、TEI コンソーシアムと新 TEI ガイドライン(P5)の概説を行う。

TEI ガイドラインは、人文科学資料を電子化する際の、中心的な道具のひとつ として、学術研究の分野における電子テキストの作成や利用形態を、根本的に 変えてきた。TEI コンソーシアムが管理している TEI ガイドラインは、現在、 電子図書館、学術出版、古典籍アーカイブ、言語資料、個人研究プロジェクト、 主題研究用コレクションなど、幅広い学術分野で利用されている。TEI コミュ ニティでは、極めて便利なテキストの符号化システムを提供している。これは、 人文科学テキストを、単純かつ高度に複雑なデータ表現形式といった多様なレ ベルにおいて、作成・蓄積・交換・保存を可能にする、効果的でかつ記述能力 の高い方法となっている。本発表では、はじめに、TEI ガイドラインを、テキ ストの標準符号化方式という視点から解説し、次に、TEI 利用した実際のプロ ジェクトにおいて、どう修正・導入されているかを解説する。各プロジェクト では、TEI に独自の修正を加えた形式を採用している。これらの各定義と TEI ガ イドライン本体との関係について解説する。また、これらのカスタマイズの実 情を見ながら、同時に、TEI ガイドライン自体の構造の解説も行う。TEI コンソー シアムの組織構成について、SIG の構成などを例に解説する。SIG とは、特定分 野に専念したグループのことである。最後に、個人、組織、団体がどう連携してゆくかについて述べたい。

国際・地域対応版 TEI にむけて

Sebastian Rahtz

今日まで、TEI ガイドラインは、ヨーロッパ、アメリカ、アジアにおけるプロ ジェクトや機関で広く採択され、様々な言語のテキストを符号化する際に利用 されてきた。しかし、TEI ガイドラインは、英語で書かれている。用例の殆ど は英文学からのもので、要素名にいたっては、略(英単)語になっている。TEI 活動やその成果である TEI ガイドラインは、より便利に世界中で使用されるためには、国際化され、地域に対応すべきであることを理解する必要がある。

本稿は、TEI がどのように国際化に対応し得るのかについて、以下の点を扱う。

- 国際化・地域化が課題となる理由
- TEI アーキテクチャーが国際対応版に向けてどう対応していくか
- W3C ITS(Internationalization Tag Set)ガイドラインに TEI はどう対応するか
- 予備実験の成果と、今後の計画
- 国際対応版 TEI を使用する際に必要なツール

伝記・人物研究情報のマークアップ

Matthew J. Driscoll

本稿では、現在取りかかっている、TEI を使用して、伝記・人物研究の情報を マークアップする作業について報告する。ここで扱っている情報とは、人に関 する情報で、例えば、生年月日、死亡日時、生誕地、死没地、婚姻情報、家族 関係、出身階級、居住地、学歴、職歴、宗教、職能、などである。

XQuery を使ってテキストを読む

James Cummings

本稿では、はじめに、W3C XQuery(XML Query Language)の概要を紹介し、 XQuery の基本と、XML データベースを使った際の可能性を、例示してゆく。 次に、XQuery にある多様な表現と機能を紹介する。この多くは XPath から継承 されたもので、XML データベースからデータを抽出したり場所を指定する際に 使用される。次に、TEI P5 に準拠した XML 文書に対して実際に XQuery を使用することを紹介する。この際、ネイティブ XML データベースとして人気のある eXist を使用し、名前空間の使用と、eXist が持つ便利な機能も紹介する。 TEI データを扱う際に、どうクエリを書くのかについてのデモを行う予定である。更に、Cocoon と eXist を使ったサーバから XQuery を使って情報検索をする簡単な Web システムをどう構築するか、順を追って説明してゆく。

トピックマップを使っての TEI テキスト

Conal Tuohy

本稿では、複雑な TEI テキストをどう表示するかについて扱う。

テキスト・アーカイブでは、多くが TEI テキストを HTML に変換し、Web 上で公開 している。その際、TEI テキストにある「章」や「ページ」は、独立した web ペー ジへと変換されている。このような手法では、物理的な書籍と同じような構造 を持った web サイトが構築されることになる。

しかし、TEI は HTML よりも強力であり、「章」「ページ」「段落」などといったものよりも、もっと他のより魅力的な素性を符号化することができる。例え ば、TEI は、文学批評や言語学的分析の他にも、人物、場所、事態といった情 報も符号化することが可能である。実際、TEI は、研究者が必要とするあらゆるものに対応できるよう拡張が可能である。

テキストの符号化をより複雑なものにすることは、データを HTML に変換することよりも、より難しくなる。TEI は、符号化する研究者にとっては、複雑な情 報も符号化できるようなデザインになっているが、その結果として、利用者に とっては理解が難しくなるかもしれない。そのため、データを表示する際には、 TEI データをより適切な別の形で表示する必要がある。例えば、TEI のデータに 人物への参照が含まれているとすると、その一覧を作成する際には、その参照 を集めることになる。実

際には、多くの場合で、後で検索したり web サイト上 に載せたりするために、TEI データから情報を抽出し、データベースに入れる 必要がある。

新しいトピックマップの ISO 規格は、これらの問題に対処するものになっている。トピックマップとは、自在な構造を持つ、一種の web データベースである。 本発表では、トピックマップを使って、TEI データから、様々な使い方ができる大規模な web サイトを作る枠組みを、デモと共に解説する。

ポスター

TEI @ RCH

Dot Porter (ケンタッキー大学)

人文科学コンピュータ共同研究施設(RCH; The Collaboratory for Research in Computing for Humanities, the University of Kentucky)では、新規開発 プロジェクトで TEI P5 を利用している。本ポスターでは、現在、RCH で行われているプロジェクトに焦点を当て、TEI P5 による自在性を享受している取組について紹介する。

Ross Scaife 教授(古典)を中心としたロマンス語資料プロジェクトでは、学 生やイギリスのラテン語研究関連機関によって、新研究や楽しむことを目的に、 主に 16 世紀から 17 世紀の様々なテキスト資料を作成している。 この資料の、特に書誌情報の引用・参照を符号化するために、中核モジュール (core)、芝居向け基礎モジュール(drama)、名前・日付向け追加モジュール (namesdates)、を使用している。

また、同じく Ross Scaife 教授を中心とするラテン語辞書プロジェクト(LLP; Latin Lexicography Project)では、web ベースのラテン語辞書を構築しており、 1880 年までをカバーする重要ないくつかのラテン語辞書類をデジタル化して統 合し、追加登録を行ってきた結果、現在では、いっそう網羅的なものになって きた。このプロジェクトでは、古典ラテン語、ロマンス語の辞書を符号化する 際に、辞書向け基礎モジュール(dictionaries)を使用している。

現在はまだ計画段階ではあるが、歴史学部の Abigail Firey 講師を中心とした Dacheriana コレクションプロジェクトでは、様々なカロリング朝の教会法を符 号化するために、批評研究向け追加モジュール (txtcrit)を使用する予定である。

Ben Withers 助教授(芸術・芸術史学科長)を中心とした古英語六書プロジェク トでは、アングロサクソン研究者や様々な専門家と共に、大英図書館所蔵、10 世紀の手書きものである、古英語絵入六書 "Claudis B.iv"のデジタル版を作成 する予定である。ここでは、写本向け追加モジュール (msdescription)を使用 し、更に、批評研究向け追加モジュール(textcrit)と一次資料向け追加モジュー ル (transc)の拡張を提案する予定である。

ベネチア本 A プロジェクトは、ホメロスマルチテキストプロジェクトの一部とし て、ハーバード大学古典ギリシャ研究センターとの共同研究として行っている。 ベネチア本 A プロジェクトは、ベネチアの国立聖マルコ図書館が所蔵する、最も 年代の古い、ホメロスの『イリアス』と注釈を含む、10

世紀ビザンチンの写本 であるベネチア本 A の、完全なる画像データベースを作成するものである。このプロジェクトでは、写本向け追加モジュール(msdescription)が使用される予定で、TEI と古典テキストサービス(CTS; Classical Text Services)プロ トコルとの連携や、TEI と METS 間における画像版テキストのマッピングについての解説も行う予定である。

バージョニングマシン

Susan Schreibman (メリーランド大学)

バージョニングマシンは、テキストの複数の版を表示し、比較することが出来 る、オープンソースのソフトウェアである。古写本の校訂版に見られる注釈のような情報にも対応した表示環境になっている。同時に、電子出版に耐えるような機能を持っており、例えば、書誌学向けの表示では、各版を順に表示する フレームが用意され、各版の画像データを操作し、横に並べたり、注釈を拡大 表示したりすることが出来る。

このバージョニングマシンは、TEI 準拠の XML テキストを表示することもできる。 テキストは、個々に(独立文書として)符号化したり、また、TEI の「資料研究 用タグ集合(TEI.textcrit)」に準拠して符号化することもできる。資料研究用 タグ集合は、構造化された機械可読なデータ形式でもって、様々な版を記述す る、最も効果的で精緻な手法になっている。バージョニングマシンは、 資料研究用タグ集合に準拠してテキストを符号化する際に、各版の画面や行を 合わせるといった機能を持っている。

本ポスターセッションでは、バージョニングマシンのデモを行う予定である。

TEI 外字モジュール

Christian Wittern (京都大学)

TEI ワーキンググループ「文字符号化」では、ユニコード外文字の表示に関するモジュールを開発してきた。現在、ユニコードは、XML で使用される標準の 文字符号化方式であり、従って、TEI の標準文字符号化方式にもなっている。 このモジュールでは、以下のような課題を扱っている。

- ユニコード中にある文字について、数多くのグリフの中から、どのような形の ものを表示されるについて、どう記述するのか
- ユニコードにない文字について、それをどのように記述するのか。

本稿では、このモジュールの使い方と応用の仕方を例示する。

CBETA 電子仏典

Christian Wittern (中華電子仏典協会)

中華電子仏典協会(CBETA; Chinese Electronic Buddhist Text Association) では、1998 年に、中国語の仏典「三蔵」全てを電子化するという、野心的な プロジェクトに取りかかった。少人数ではあるが、精力的に活動を行ってきた 結果、この 8 年の間に、100 巻にも及ぶ、12 億文字を超えるデータが、出版向け の PDF から、携帯電話でも読めるテキスト形式といったデータ形式で、ネット 上や CDROM を介して、無料で公開されてきた。

元テキストからの、極めて正確な書き起こしや、多くのミスプリを正し、注釈 を加えるといった、膨大な作業は、世界中の仏教研究者から、高い賞賛を得る ことになった。技術的には、高度にカスタマイズされた TEI P4 がプロジェクト 内で使用されている。しかし、現在の試験版では、TEI P5 が使用されている。 TEI P5 の外字モジュールを使用することで、この試験版では、ユニコードでは まだ規定されていない 8000 字以上の文字(異字体)を符号化している。

本ポスター発表では、これらのテキストを表示し、研究で使用するアプリケー ションを紹介する。

阿含経の解読--デジタル版『別訳雑阿含経』--

Marcus Bingenheimer (中華佛學研究所)

デジタル版『別訳雑阿含経』(BZA)は、蔣經國基金會から 3 年間の研究補助金を受け、 中華佛學研究所(台湾)が行っているプロジェクトである。

『別訳雑阿含経』は、364 小経を含む 16 巻から成る、初期漢訳仏典、阿含経に属 するものである。阿含経は、最も古い仏教経典である。その原典(サンスクリッ ト語)は、殆どは失われてしまったが、いくらかの断片が伝えられている。 この漢訳版よりさらに重要なのが南伝仏教版で、これはパーリ語で書かれた全ての 経典を含んでいる。ニカーヤ(Nikāya、阿含経の別の名称)は、英語、日本語、ドイツ語に翻訳さ れ、よく研究されているが、漢訳版や校訂版は殆ど無い。 基本的には、阿含経の 364 小経の全てに、少なくとも、中国語の一つの異訳とパーリ語の対象（コメント付で含まれている）がある。 多くの場合は中国語とパーリ語の対訳は複数であるが、サンスクリット語で書かれた断片 が現存することはまれである。

本プロジェクトの目標は、阿含経の完全デジタ ル版を作成することである。これにより、テキストのクラスターを明らかにすることが可能になる。このデジタル版は、無料で使用することができる予定で ある。本プロジェクトでは、阿含経の英語版も作成している。漢訳テキストには、 中華電子仏典協会(CBETA; Chinese Electronic Buddhist Text Association) 版を提供し、パーリ語版は、ヴィパッサナー研究所(Vipassana Research Institute)から Chaṭṭha Saṅgāyana CD のテキストを使用して良いと の許可を頂いている。

XML によるマークアップは、TEI に準拠して行われ、一般ユーザには、HTML 形式 でデータを公開している。マークアップによって、対話的な部分、名前、韻文 部と散文部の違い、典拠が明確な版との関連性などを、構造的に記述すること が出来る。マークアップによって、パーリ語部分と漢訳部分との対応を関連付 けることが可能になる。

クラスター中のテキストは、比較カタログによって、関連づけることが可能になる。時間があれば、より細粒に関連を付けれる為にテキストクラスターの中身をさらに細かくマークアップを行う予定である。 ソースファイルとユーザアプリケーションのミドルウェアとして、XML データベース である eXist を使用している。eXist を使ったデータサービスシステムは、仏 教研究だけではなく、台湾の人文科学研究においても、初めてのケースである。 ユーザは、ネット上から、必要なクラスターを選択し、その中にあるテキスト を、3 つの画面メニューで、表示することが可能である。

本デジタル版では、以下のことが可能である。

- クラスター内で、複数のテキストを比較することができる。
- 364 小経(クラスター)の対照と内容を確定、追加すること。
- 『別訳雑阿含経』と雑阿含経に、新しい訓点を追加することができる。
- 『別訳雑阿含経』の特定節に、英語訳を付記することができる。
- 対訳のコーパスを作成し、統計分析を可能になる。
- 将来、資料を追加することができる。
- 将来、阿含経を再構成する際のモデルとなる。

テキスト大海の航海法 -- トピックマップと A.C.スウィンバーンの詩--

John Walsh and Michelle Dalmau (インディアナ大学**)**

XML トピックマップ(XTM)は、XML によるトピックマップの事で、メタデータを 扱う、強力で自在なデータフォーマットである。 XTM には、デジタル資料のインタフェースや、TEI 準拠の巨大テキスト資料といっ た、人文科学研究向け資料から情報を発見する新しい仕組みとなる可能性がある。 トピックマップ支持者は、トピックマップによる構造化によって、情報検索は 劇的に改善すると主張しているにもかかわらず、人文科学研究者が、トピック マップがもたらす豊かで自在な概念関係から、どれ程の恩恵が得られるものか を明らかにするといったユーザ主体の活動は、殆どみられない。

本ポスターでは、トピックマップの紹介を行い、TEI 準拠のテキストコレクショ ンをどう使用し、どのような恩恵が得られるかについて、スウィンバーンプロ ジェクトを例に紹介する。 また、本ポスターでは、トピックマップベースと一般的な検索ベースのインタ フェースとで使用感を比較する手法についても解説する。 この比較研究調査による主な結果を基に、インタフェースのデモを行う。 また、この調査結果を基にした、別のデザインも紹介する。

本研究の目的は、人文科学のデジタル化研究において、トピックマップの論議 を活発化することにあり、人文科学研究者が、発見過程の中で普段使用している分類構造を超えたものを、トピックマップがもたらすことを示したい。 また、人文科学資料のデジタル化を、トピックマップやこれと似たメタ

データ 構造を使って検討している人に、成功するユーザインタフェースのデザインを 示したいと考えている。

XXQ: XML 資料向けクエリ言語

Lou Burnard (オックスフォード大学)

本ポスターでは、XXQ の紹介を行う。XXQ とは、新しい XML 向けクエリ言語で、 Xaira(テキスト検索エンジン)と共に使用されることを前提に、現在、開発が 進められているものである。本ポスターでは、Xaira の紹介も行うが、発表の 中心は、検索エンジン独立の XML 向けクエリ言語のアイディアを紹介すること にある。XXQ は、"XPath+"と呼べるものかもしれない。XXQ の主な特徴は、以下である。

- XQuery とは異なり、XML 構造が関係データ構造であるようには振る舞わない。
- XPath とは異なり、単一構造を前提とした検索という制限は設けない。
- grep とは異なり、検索対象は、文字ではなく語彙トークンになる。

XXQ は、正規表現の記述力を持つパターンマッチング言語で、(先読みしないと いう)比較時の弱点はあるものの、簡単な XML で記述できるものである。

Syntax から見たマークアップの課題

大矢一志 (鶴見大学)

マークアップする際に困難と感じる原因のうち、ML の Syntax から見た「ひっか かりどころ」を紹介する。特に、はじめて本格的なマークアップによるデータ 作成を試みようとしたり、これから TEI 準拠のデータ作成を試みられているケー スで必要であろう検討基準を紹介したい。

TEI ガイドライン日本語版プロジェクト

大矢一志 (鶴見大学)

Christian Wittern (京都大学)

日本におけるマークアップ資料の作成を支援するために、TEI ガイドラ イン日本語版を作る計画があり、現在、P4 を土台に翻訳が進められている。 目標としては、P5 の日本語版を作成する予定である。計画の概要と進行状況を 紹介し、参加者を募りたい。

『資治通鑑』マークアップ

中楯はまな (京都大学人文科学研究所)

「21 世紀 COE 東アジア世界の人文情報學研究教育據點」内の「漢字文献ナリッジベースの構築」班では、中国の代表的な編年体の歴史書である『資治通鑑』の 中から唐紀（618-906）部分を中心として、そこから得られる人物、地域、著作 物などの情報を検索、利用できるためのマークアップを行っている。 ここまでのマークアップの進捗状況と、実例、課題、可能性を挙げたい。 漢字文献研究に関する伝統的な知識と最先端のデジタル化技術の融合の中で、21 世紀における新しい東アジア学のあり方、漢字文化の記述の方法を追求する 試みのひとつとして紹介したい。